

**Final Report**

**to**

**Sandia National Laboratory  
Minerals Management Service**

**RAYLEIGH WAVES IN OFFSHORE ENVIRONMENT**

**by**

**Toyoaki Nogami and Akira Igarashi**

**Scripps Institution of Oceanography  
University of California, San Diego  
La Jolla, California 92093**

## TABLE OF CONTENTS

	page
INTRODUCTION	1
MATHEMATICAL FORMULATION	2
1. Field Equation	3
2. Finite Element Formulation	4
NUMERICAL STUDIES	9
1. Rayleigh Wave Generation	9
2. Conditions Considered	10
3. Numerical Results	11
CONCLUSIONS	14
FUTURE STUDY	16
REFERENCES	17
APPENDIX A: DERIVATION OF GROUP VELOCITIES	A1
APPENDIX B: COMPUTER PROGRAM RLW	B1
APPENDIX C: LISTING OF COMPUTER PROGRAM RLW	C1

## INTRODUCTION

In an offshore environment, soil deposits are submerged under a certain depth in the water. If this environment affects the seismic response of the soil deposits, the onshore seismic ground response records can not be used directly for the seismic response analysis of offshore structures. However, the seismic offshore ground motion records are scarce and those recorded in onshore are often used directly without consideration of their applicability to the offshore environment.

A combination of the S-waves and P-waves forms the horizontally propagating Rayleigh waves to satisfy the homogeneous boundary conditions of the soil medium. Since the Rayleigh waves are partly formed by the P-waves, the water above and inside of the soil deposits may be strongly coupled with the soil skeleton to modify the soil motions. The Rayleigh waves are long period waves and often significant in long period components of seismic ground motions. Since the fundamental periods of pile-supported offshore platforms are relatively long, they may be excited significantly by the Rayleigh waves. Therefore, it is very important to study the offshore environment effects on those waves. As the first step to understand the difference between the offshore and onshore seismic motions, the offshore environment effects are investigated for individual mode of Rayleigh waves in this study. They are also investigated briefly for the body waves as a special case.

## MATHEMATICAL FORMULATION

### 1. Field Equation

The equations of motion for small strains of a saturated porous media are provided by Biot (1). The porous media can be viewed as a mixture of solid-phase skeleton and fluid-phase pore. A dynamic equilibrium equation can be written for solid-fluid phases as:

$$L^T \sigma + \rho b - \rho \ddot{u} - \rho_f \dot{w} = 0 \quad (1)$$

and also a generalized Darcy law for the dynamic equilibrium of the fluid is expressed as

$$\nabla \pi + \rho_f b - k^{-1} \dot{w} - \rho_f \ddot{u} - (\rho_f/n) \dot{w} = 0 \quad (2)$$

where a superposed dot indicates a time derivative,  $\sigma$  = total stress vector,  $u$  and  $w$  = displacement vectors containing the displacements of solid-phase and displacements of fluid-phase relative to the solid phase, respectively,  $n$  = porosity,  $k = k_0/(\rho_f g)$  with  $k_0$  = permeability coefficient and  $g$  = gravity,  $\rho$  and  $\rho_f$  = densities of fluid-solid mixture and fluid, respectively,  $b$  = unit volume body force vector,  $\pi$  = pore fluid pressure ( $\pi > 0$  for tension),  $\nabla^T = (\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3)$ , and

$$L^T = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_3} \\ 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix} \quad (3)$$

It is noted that the relative fluid displacement,  $w$ , is defined as  $w = n(V - u)$ , where  $V_i$  is the displacement field for fluid defined so that the volume of fluid displaced through a unit area perpendicular to  $x_i$  axis is  $nV_i$ . The total stress and density of the mixture are

$$\begin{aligned}\sigma &= (1-n)\sigma_s + n\pi m \\ \rho &= (1-n)\rho_s + n\rho_f\end{aligned}\tag{4}$$

where  $\rho_s$  = density of soil material,  $\sigma_s$  = solid phase stress vector and  $m^T = (1, 1, 1, 0, 0, 0)$ .

The stress-strain relationship of a linear isotropic elastic material is

$$\sigma = De + \alpha m\pi\tag{5a}$$

or

$$\begin{aligned}\sigma &= (D + \alpha^2 Qmm^T)e + \alpha Qm\zeta \\ &= (D + \alpha^2 Qmm^T)Lu + \alpha Qm\nabla^Tw\end{aligned}\tag{5b}$$

and the constitutive relationship for the fluid phase is

$$\begin{aligned}\pi &= \alpha Qm^Te + Q\zeta \\ &= \alpha Qm^TLu + Q\nabla^Tw\end{aligned}\tag{6}$$

where  $e (= Lu)$  and  $\zeta (= \nabla^Tw)$  are strains in the solid and the volumetric strain in the fluid respectively;  $D$  = drained material stiffness matrix; and  $\alpha$  and  $Q$  = parameters explained later. According to Simon et al. (3,4), the following relations are obtained:

$$\alpha = 1 - \frac{K_d}{K_s} \quad (7)$$

$$\frac{1}{Q} = \frac{1}{K_f} + \frac{\alpha - n}{K_s}$$

where  $K_s$  = bulk modulus relating pore fluid pressure to volumetric strain in the solid skeleton (e.g.  $\pi = K_s m^T e$ );  $K_d$  = bulk modulus of drained solid skeleton =  $E/(3(1-2v))$  with drained elastic constants  $E$  and  $v$ ; and  $K_f$  = bulk modulus of the fluid.

Substitutions of Eq. 5b into Eq. 1 and of Eq. 6 into Eq. 2 result in

$$\begin{aligned} L^T(D + \alpha^2 Q mm^T) Lu + L^T \alpha Q m \nabla^T w + \rho b - \rho \ddot{u} - \rho_f \dot{w} &= 0 \\ \nabla \alpha Q m^T Lu + \nabla Q \nabla^T w + \rho_f b - k^{-1} \dot{w} - \rho_f \ddot{u} - (\rho_f/n) \dot{w} &= 0 \end{aligned} \quad (8)$$

## 2. Finite Element Formulation

The medium is divided into a number of horizontal homogeneous layers and the coordinate system is assigned as shown in Fig. 1. Plane strain conditions,  $du/dz = dw/dz = 0$ , are considered. Adopting the interpolations scheme in  $y$  direction and assuming waves propagating in the positive  $x$  direction, displacements of a fluid saturated layer of thickness  $H$  are expressed as

$$\begin{aligned} u(x,y) &= N_a(y) U_1 e^{-ihx} + N_b(y) U_2 e^{ihx} \\ w(x,y) &= N_a(y) W_1 e^{ihx} + N_b(y) W_2 e^{ihx} \end{aligned} \quad (9)$$

or in a more compact form

$$\begin{aligned} \mathbf{u}(x,y) &= \mathbf{N}(y) \mathbf{U} e^{-ihx} \\ \mathbf{w}(x,y) &= \mathbf{N}(y) \mathbf{W} e^{-ihx} \end{aligned} \quad (10)$$

where  $h$  = wave number;  $i = -1$ ;  $\mathbf{U}_1 e^{-ihx}$  and  $\mathbf{U}_2 e^{-ihx}$  = solid-phase displacement vectors at the top and bottom end of the layer, respectively;  $\mathbf{W}_1 e^{-ihx}$  and  $\mathbf{W}_2 e^{-ihx}$  = fluid-phase relative displacement vectors at the top and bottom ends of the layer, respectively;  $\mathbf{U}^T = (\mathbf{U}_1^T \ \mathbf{U}_2^T)$ ;  $\mathbf{W}^T = (\mathbf{W}_1^T \ \mathbf{W}_2^T)$ ; and

$$\mathbf{N}(y) = \begin{bmatrix} N_a(y) & 0 & N_b(y) & 0 \\ 0 & N_a(y) & 0 & N_b(y) \end{bmatrix} \quad (11)$$

$$N_a(y) = 1 - \frac{y}{H} \quad \text{and} \quad N_b(y) = \frac{y}{H}$$

Consider a layer subjected to the vertical and horizontal tractions at the top end,  $\mathbf{p}_1(x) = P_1 e^{-ihx}$ , and the tractions at the bottom end,  $\mathbf{p}_2(x) = P_2 e^{-ihx}$ . Then, the Galerkin procedure results in

$$\int_0^H \mathbf{N}^T \mathbf{F}(x,y) dy = \mathbf{p}(x) \quad (12)$$

where  $\mathbf{F}(x,y)$  = equilibrium equation expressed as  $\mathbf{F}(x,y) = 0$ ; and  $\mathbf{p}(x)^T = (\mathbf{p}_1(x)^T \ \mathbf{p}_2(x)^T)$ . Substitutions of Eq. 10 into Eq. 8 and then Eq. 8 into Eq. 12 lead to

$$\begin{aligned}
& \int_0^H N^T L^T (D + \alpha^2 Q m m^T) L N e^{-ihx} U dy + \int_0^H N^T L^T \alpha Q m \nabla^T N e^{-ihx} W dy - \\
& \int_0^H N^T \rho N e^{-ihx} \dot{U} dy - \int_0^H N^T \rho_f N e^{-ihx} \dot{W} dy = p_u e^{-ihx} \\
& \int_0^H N^T \nabla \alpha Q m^T L N e^{-ihx} U dy + \int_0^H N^T \nabla Q \nabla^T N e^{-ihx} W dy - \int_0^H N^T k^{-1} N e^{-ihx} W dy - \\
& \int_0^H N^T \rho_f N e^{-ihx} \dot{U} dy - \int_0^H N^T \frac{\rho_f}{n} N e^{-ihx} \dot{W} dy = p_w e^{-ihx} \quad (13)
\end{aligned}$$

After performing a partial differentiation with respect to  $x$  required in  $L$  and  $\nabla$  and dividing the left- and right-hand sides of the equation by  $e^{-ihx}$ , two equations in Eq. 13 are rewritten in a single matrix equation expressed by

$$\begin{bmatrix} m_{uu} & m_{uw} \\ m_{wu} & m_{ww} \end{bmatrix} \begin{Bmatrix} \dot{U} \\ \dot{W} \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & c_{ww} \end{bmatrix} \begin{Bmatrix} \dot{U} \\ \dot{W} \end{Bmatrix} + \begin{bmatrix} k_{uu} & k_{uw} \\ k_{wu} & k_{ww} \end{bmatrix} \begin{Bmatrix} U \\ W \end{Bmatrix} = \begin{Bmatrix} P_u \\ P_w \end{Bmatrix} \quad (14)$$

where the first, second and third matrices are mass, damping and stiffness matrices, respectively. The inside terms of those matrices are defined by

$$\begin{aligned}
m_{uu} &= \rho \int_0^H N^T N dy & m_{uw} = m_{wu} &= \rho_f \int_0^H N^T N dy & m_{ww} &= \frac{\rho_f}{n} \int_0^H N^T N dy \\
c_{ww} &= k^{-1} \int_0^H N^T N dy \\
k_{uu} &= \int_0^H B_u^T (D + \alpha^2 Q m m^T) B_u dy & k_{uw} = k_{wu}^T &= \alpha Q \int_0^H B_u^T m B_w dy \\
k_{ww} &= Q \int_0^H B_w^T B_w dy
\end{aligned} \tag{15}$$

where

$$B_u = \hat{L}N \quad \text{and} \quad B_w = \hat{\nabla}N \tag{16}$$

with

$$\begin{aligned}
\hat{L}^T &= \begin{bmatrix} -ih & 0 & \frac{d}{dy} \\ 0 & \frac{d}{dy} & -ih \end{bmatrix} \\
\hat{\nabla}^T &= (-ih, \frac{d}{dy})
\end{aligned} \tag{17}$$

The results of evaluation of Eq. 15 are given in Tables 1 through 3.

Following the standard finite element method procedure, the finite element equation of motion of a water saturated layered medium can be formulated after superimposing the matrices of all layers. This results in

$$\dot{\mathbf{M}}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{P} \quad (18)$$

where  $\mathbf{M}$ ,  $\mathbf{C}$  and  $\mathbf{K}$  = global mass, damping and stiffness matrices obtained after superimposing those of layer matrices, respectively;  $\mathbf{U}^T = (\mathbf{U}^T \mathbf{W}^T)$ ;  $\mathbf{P}^T = (\mathbf{P}_u^T \mathbf{P}_w^T)$ . The Rayleigh waves can be defined from the homogeneous boundary conditions ( $\mathbf{P} = \mathbf{0}$  in Eq. 18) and a harmonic motion field, and thus by solving

$$[-\omega^2 \mathbf{M}(h) + i\omega \mathbf{C}(h) + \mathbf{K}(h)]\mathbf{U} = \mathbf{0} \quad (19)$$

where  $\omega$  = circular frequency. Expressions of  $\mathbf{M}$ ,  $\mathbf{C}$  and  $\mathbf{K}$  given in Tables 1 through 3 enable Eq. 19 to be rewritten as

$$[h^2 \mathbf{E}(\omega) + ih\mathbf{F}(\omega) + \mathbf{G}(\omega)] \mathbf{U} = \mathbf{0} \quad (20)$$

Eigenvalues ( $\omega$  or  $h$ ) and eigenvectors ( $\mathbf{U}$ ) are determined by either Eq. 19 for given  $h$  or Eq. 20 for given  $\omega$ .

Phase velocities of the Rayleigh waves are

$$V_{ph} = \frac{\omega}{h} \quad (21)$$

According to Lysmer (2), group velocities of the Rayleigh waves are

$$V_g = \frac{d\omega}{dh} = \frac{\mathbf{U}^T \frac{\partial}{\partial h} \mathbf{K} \mathbf{U}}{\mathbf{U}^T [2\omega \mathbf{M} - i \mathbf{C}] \mathbf{U}} \quad (22)$$

The derivation of Eq. 22 is given in Appendix A.

## NUMERICAL STUDIES

### 1. Rayleigh Wave Generation

For a given frequency, wave numbers determined by solving Eq. 20 are complex numbers in general. All of those waves are called as the generalized Rayleigh waves. Only those with non-zero real part of wave number can propagate in  $x$  direction and are recognized at distances far from the seismic source. When the wave number consists of both real and imaginary parts, those propagating waves decay with the distance  $x$  and the rate of decay is proportional to the value in the imaginary part. On the other hand, the generalized Rayleigh waves with pure imaginary wave number do not form wave motions in the  $x$  direction.

For a given frequency, the wave numbers and their mode shapes can be determined by solving Eq. 20 for a descritized horizontally layered medium. Thus, superimposing those mode shapes, the displacements of a free-field soil deposit along the depth can be expressed as

$$U(x, \omega) = \sum_{j=1} \phi_j(\omega) e^{-ih_j x} \eta_j \quad (23a)$$

or in a more compact form

$$U(x, \omega) = \Phi(x, \omega) \eta \quad (23b)$$

where  $h_j$  and  $\phi_j(\omega)$  = jth wave number and its associated Rayleigh wave mode shape vector, respectively;  $\eta_j$  = jth generalized coordinate; and  $\phi_j(\omega) e^{-ih_j x}$  occupies the jth column in  $\Phi(x, \omega)$ .

It is assumed that  $U(x_0, \omega)$  is the  $\omega$ -frequency component of the displacement vector  $U(x_0, t)$ , at a certain lateral distance  $x_0$ . Such  $U(x_0, t)$  may be determined by the analysis of near-field motions induced by a specific seismic source. Then, Eq. 23b results in the following equation to define  $\eta_j$ :

$$U(x_0, \omega) = \Phi(x_0, \omega) \eta \quad (24)$$

After determining  $\eta$  by solving Eq. 24 for various frequency components, the Rayleigh wave motions at the distance  $x$  can be computed by Eq. 23a or 23b for those frequencies. Those motions are related to the time-domain motions through a Fourier inverse.

## 2. Conditions Considered

Soil deposits in an offshore environment are submerged and located under a certain depth in the sea water. In order to look at the effects of such an environment, wave modes and dispersion curves of Rayleigh waves were computed for the three cases as shown in Fig. 2, including those with air over dry soil deposits (air-solid), air over submerged soil deposits (air-mixture) and water over submerged soil deposits (water-mixture). The water depth is assumed to be 3/5 of the thickness of the soil deposits.

Conditions of soil frame and water considered herein are given in Fig. 3, in which Profile A and Profile C are homogeneous profiles and Profile B is an inhomogeneous profile. The shear modulus of the inhomogeneous profile (Profile B) increases in a step-wise with depth. Profile A and Profile C are identical except  $n$  and  $k$  smaller in Profile C than Profile A. The soil deposits and water are divided respectively into ten and three equal thickness horizontal layers. In this study, Eq. 19 was used by prefixing the wave number  $h$ .

### 3. Numerical Results

The Rayleigh wave dispersion curves, shown in Figs. 4a through 4c, are constructed by calculating the frequencies at various prefixed wave numbers for soil profiles considered. In each profile, completely dry (air-solid), air over submerged soil (air-mixture) and water over submerged soil (water-mixture) are considered. Elastic constants of the solid frame is assumed to be identical between the dry and submerged conditions for the convenience of the study of the effects of the pore fluid in the soil mass. The shear wave velocity,  $v_s$ , equal to 228.84 m/sec is used to normalize the frequency for Profile B (inhomogeneous profile): this corresponds to the average shear wave velocity of Profile B. The slopes of the curves in the figure are the ratios, Rayleigh wave phase velocity over shear wave velocity. When the wave number is zero, the body waves propagates in the straight vertical direction. Therefore, the frequencies at  $h = 0$  in Figs. 4a through 4c are the natural frequencies of the soil deposits under one dimensional body waves vertically propagating in soil deposits. Those frequencies for a homogeneous solid stratum are  $\omega H/v = j\pi/2$  ( $j = 1, 3, 5, 7, \dots$ ) in which  $v$  is either S-wave velocity ( $v_s$ ) for S-waves or P-wave velocity ( $v_p$ ) for P-waves.

Let us first look at the dispersion curves for the dry soil case. The behaviors associated with each of the dispersion curves at  $hH = 0$  are those of one particular body in waves vertically propagating. For example, the curves of the first and third mode waves in Fig. 4a intercept respectively at  $\omega H/v_s = \pi/2$  and  $3\pi/2$  and therefore those waves are S-waves, whereas the curves of the second and fourth mode waves intercept respectively at  $\omega H/v_p = \pi/2$  and  $3\pi/2$  and therefore those waves are P-waves. Tangent slopes of the curves are zero at  $h = 0$  and increase with  $hH$  to one and  $v_p/v_s$  respectively, for the first three mode waves and the fourth mode wave. This indicates that the Raleigh waves at sufficiently large  $h$  exhibit predominantly the S-wave characteristics in the first three mode waves and the P-wave characteristics in the fourth mode wave in the range of sufficiently large  $hH$ . This is because, as increasing  $hH$ , the ray path orientations of the body waves forming the Rayleigh waves approach to the horizontal direction thus resulted in the phase velocities approaching to the predominant body wave velocities. Since there is no imaginary part in the complex frequency, the waves are not damped at all in the dry soil case.

When the soil frame is submerged, the motions are induced in the fluid as well as the solid frame and the damping develops because of the relative motions between the soil frame and pore fluid. Thus, the imaginary part of the frequency appears in the dispersion curves. Two types behaviors can be seen in Figs. 4a, 4b and 4c in the submerged soil. One type shows the real part behavior similar to that observed for dry soil and thus is associated with significant motions in the soil frame. On the other hand, the complex frequency in the second type consists of no real part but only the imaginary part at  $h = 0$ : this indicates that the wave motion does not exist in this type and the motion decays monotonically with time. Those curves of the second type increase very sharply with  $hH$  so that the slopes in real part of dispersion curves are between those of  $v_p/v_s$  and  $v_f/v_s$ , in which  $v_f$  = wave velocity of acoustic wave in water ( $= (K_f/\rho_f)^{0.5}$ ). Significant fluid

motions are found to be associated with this second type behavior, after a close examination of the displacements in the solid frame and fluid. The second type behavior eventually becomes the first type behavior as the the wave number becomes sufficiently large except for the first mode wave in the air-mixture case as shown later . The imaginary part or damping appears to increase as the soil permeability decreases. As is seen in Figs. 4a through 4c, the water in the soil pore and above the soil surface significantly affects the dispersion curves in general.except  $hH = 0$  whereas it does very little at  $hH = 0$ .

Normalized displacement shapes along the depth are shown in Fig. 5 for  $hH= 0$ . The normalization is made by dividing the displacements by the largest one between the x and y componets of soil frame displacement amplitude at the soil surface. The displacement of the fluid is defined as the relative displacement in the pore space ( $=w/n$ ). The values are not plotted when they are very small. The soil frame displacements for  $hH = 0$  are pure horizontal in the first and third mode waves and pure vertical in the second and fourth mode waves as are seen in Fig. 5. This can be understood because the waves are respectively the vertically propagating SH-waves and P-waves in those two cases. Relative fluid displacements in the x direction are nearly identical to the x direction solid frame displacement but in the opposite direction as shown in Figs. 5a and 5b. Therefore the absolute fluid displacements are negligible when the motions are in the x direction at  $hH = 0$ . On the other hand, the absolute fluid motion is induced in the y direction motions at  $hH = 0$ . The soil frame motions are very little or not affected by the pore fluid when  $hH = 0$ .

Fig. 6 shows normalized displacements along the depth for  $hH = 5$ , computed in a similar manner as that in Fig. 4. Because of a strong coupling between the P-waves and S-waves in forming Rayleigh waves at  $hH = 5$ , the motions now have both in x and y components. The phase shifts between the two component motions are nearly  $90^\circ$  although the damping caused by the fluid motions tends to distort it. It is interesting to

note that the fluid motions in the first mode wave at  $hH = 5$  are significant for the air-mixture case but does not for the water-mixture case. This implies that very flat dispersion curves ( $v_{ph} \sim 0$ ) correspond to waves causing significant fluid motions. For a particular case considered, the reduction in porosity (and thus permeability) increases the fluid motions as shown in Fig. 6c and therefore increases the damping as noticed after comparing between the imaginary parts of the frequencies computed for two porosities (e.g. Fig. 4b (b) and Fig. 4c (b)). In general, the pore water and water above the soil surface affect very little the soil motions when the body waves are limited to propagate only in a straight vertical direction ( $hH = 0$ ) but affect significantly for other cases.

Soil surface displacement amplitude ratios between the two components of the soil frame motions, associated with an individual Rayleigh wave, are plotted in Fig. 7 for various  $hH$ . The ratio is computed by dividing the minor component by the major component. The major component is designated in the figure by either  $x$  for the  $x$  component or  $y$  for the  $y$  component. The effects of soil pore water and water above the soil surface are complex and no clear trends can be observed in this figure.

## CONCLUSIONS

The effects of an offshore environment are investigated on the Rayleigh waves motions, including the vertically propagating one dimensional body waves as a special case. Offshore soil deposits are considered to be submerged under a certain depth of the water. Analytical formulations of surface wave motions are developed by the finite element scheme taking into account a coupling between the fluid and soil frame motions. Numerical analyses are conducted for dry soil deposits (air-solid), air over the submerged

soil deposits (air-mixture) and water over the submerged soil deposits (water-mixture) assuming three different soil profiles. The computed results for those different cases have revealed the followings:

1. The offshore environment affects very little the soil motions associated with vertically propagating body waves but does significantly the Rayleigh wave motions in general.
2. Relative motions between the solid frame and the pore water produce a damping under the submerged condition, and this trend appears to be more significant for the smaller porosity (or permeability) case considered herein.
3. The real parts start at zero and sharply increase with  $hH$ , in some of Rayleigh wave dispersion curves in submerged soils, and significant fluid motions are associated with this region of the behavior.
4. For the air-mixture case, the real part of the dispersion curve corresponding to the fundamental mode starts at zero and becomes flat with increasing the wave number. Significant fluid motions are associated with this wave mode.
5. The offshore environment can both increase and decrease the vertical seafloor motion relative to the lateral motion, and no clear trends can be observed.

## FUTURE STUDY

The present study clearly shows that the offshore environment affects the Rayleigh wave soil motions. The study was made for an individual wave mode independently although the motions are resultants of the superposition of the motions associated with various modes as indicated in Eq. 23a: significance of an individual mode is defined by the value  $\eta_j$  in Eq. 23a. In addition, since the offshore environment produces various degrees of damping among mode waves, some mode waves may be damped out with the distance quickly others may slowly. Therefore, the offshore environment needs to be further investigated for  $U$  defined in Eq. 23 in order to draw a more complete picture.

Furthermore, since the offshore environment affects the Rayleigh waves definitely, it is important to further investigate how the long periods portion of the seismic response spectra are affected by the offshore environment. In those studies,  $U$  defined in Eq. 23 will be computed for various cases by the procedure described above and below Eq. 24.

## REFERENCES

1. Biot, M. A., "Mechanics of Deformation and Acoustic Propagation in Porous Media," Journal of Applied Physics, Vol. 33, 1962, pp. 1482- .
2. Lysmer, J., "Lumped Mass Method for Rayleigh Waves," Bulletin of the Seismological Society of America, Vol. 60, No. 1, 1970, pp. 89-104.
3. Simon, B. R., Wu, J. S.-S., Zienkiewicz, O. C. and Paul, D. K., "Evaluation of u-w and u- p Finite Element Methods for the Dynamic Response of Saturated Porous Media Using One-Dimensional Model," International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 10, 1986, pp. 461-482.
4. Simon, B.R., Zienkiewicz, O. C. and Paul, D. K., "An Analytical Solution for the Transient Response of Saturated Porous Elastic Solids," International Journal for Numerical and Analytical Methods in Geomechanics, Vol. 8, 1984, pp. 381-398.
5. Zienkiewics, O. C. Chang, C. T. and Bettess, P., "Drained, Undrained, Consolidation and Dynamic Behavior Assumptions in Soils," Geotechnique, Vol. 30, No. 4, 1980, pp. 358-395.

TABLE 1  
EXPRESSIONS OF SUB-MATRICES IN M

$$m_{uu} = \frac{\rho H}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ & 2 & 0 & 1 \\ & & 2 & 0 \\ & & & 2 \end{bmatrix}_{\text{sym}}$$

$$m_{uw} = m_{wu}^T = \frac{\rho_f H}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ & 2 & 0 & 1 \\ & & 2 & 0 \\ & & & 2 \end{bmatrix}_{\text{sym}}$$

$$m_{ww} = \frac{\rho_f H}{6n} \begin{bmatrix} 2 & 0 & 1 & 0 \\ & 2 & 0 & 1 \\ & & 2 & 0 \\ & & & 2 \end{bmatrix}_{\text{sym}}$$

TABLE 2  
EXPRESSIONS OF SUB-MATRICES IN C

$$c_{ww} = \frac{H}{6k} \begin{bmatrix} 2 & 0 & 1 & 0 \\ & 2 & 0 & 1 \\ & & 2 & 0 \\ & & & 2 \end{bmatrix}_{\text{sym}}$$

TABLE 3  
EXPRESSIONS OF SUB-MATRICES IN K

$$k_{uu} = \begin{bmatrix} \frac{AHh^2}{3} + \frac{G}{H} & 0 & \frac{AHh^2}{6} - \frac{G}{H} & 0 \\ 0 & \frac{GHh^2}{3} + \frac{A}{H} & 0 & \frac{GHh^2}{6} - \frac{A}{H} \\ \text{sym.} & \frac{AHh^2}{3} + \frac{G}{H} & 0 & \frac{GHh^2}{3} + \frac{A}{H} \end{bmatrix} +$$

$$\frac{i\hbar}{2} \begin{bmatrix} 0 & -(A-G) & 0 & (A+G) \\ (A-G) & 0 & (A+G) & 0 \\ 0 & -(A+G) & 0 & (A-G) \\ -(A+G) & 0 & -(A-G) & 0 \end{bmatrix}$$

where  $A = \lambda + 2G + \alpha^2 Q$ .

$$k_{uw} = k_{wu}^T = \frac{h\alpha QH}{6} \begin{bmatrix} 2h & \frac{i3}{H} & h & \frac{i3}{H} \\ \frac{i3}{H} & 0 & \frac{i3}{H} & 0 \\ h & \frac{i3}{H} & 2h & \frac{i3}{H} \\ \frac{i3}{H} & 0 & \frac{i3}{H} & 0 \end{bmatrix} +$$

$$\frac{\alpha Q}{H} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

$$k_{ww} = \frac{hQH}{6} \begin{bmatrix} 2h & \frac{i3}{H} & h & \frac{i3}{H} \\ \frac{i3}{H} & 0 & \frac{i3}{H} & 0 \\ h & \frac{-i3}{H} & 2h & \frac{i3}{H} \\ \frac{-i3}{H} & 0 & \frac{-i3}{H} & 0 \end{bmatrix} +$$

$$\frac{Q}{H} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

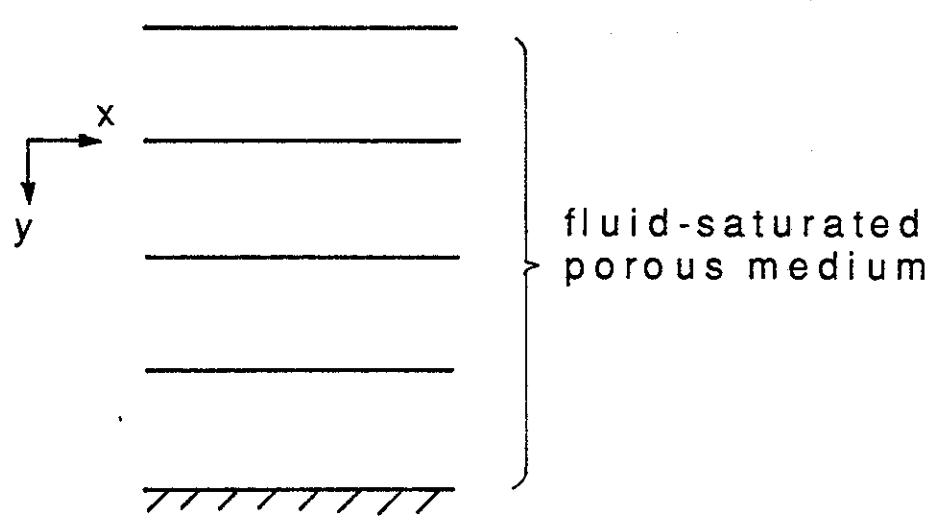


Fig. 1 Fluid-saturated porous medium divided into homogeneous layers

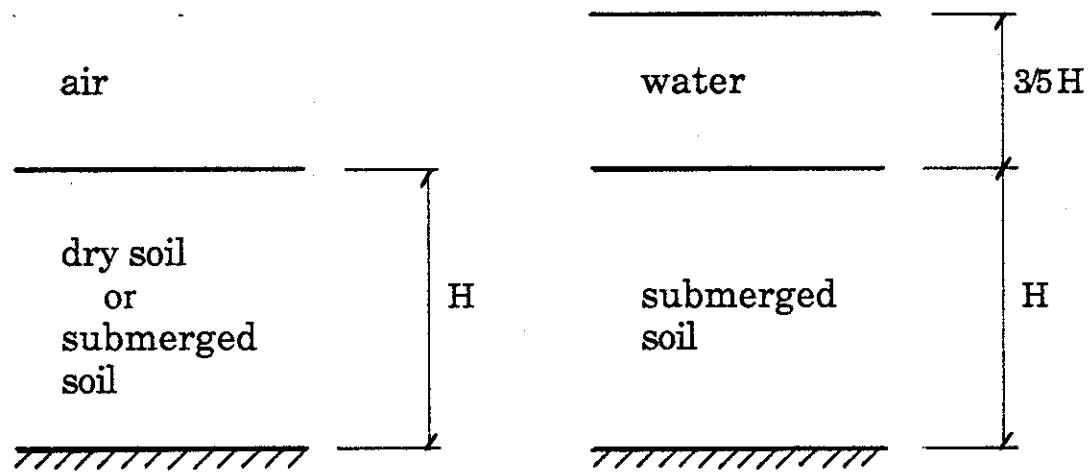


Fig. 2 Air or water above the soil deposits

**soil frame**

$$v = 0.25$$

$$\gamma_s = \rho_s g = 2000 \text{ kg/m}^3$$

$$K_s = \infty$$

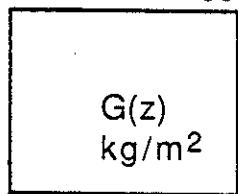
**fluid**

$$\gamma_f = \rho_f g = 1000 \text{ kg/m}^3$$

$$K_f = 2208000000 \text{ kg/m}^2$$

**Profile A**

5343648

**Profile B**

2599200

4495715

5814050

G(z)  
kg/m<sup>2</sup>

6876751

7795608

$$n = 0.5$$

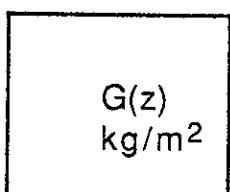
$$k = 0.001 \text{ m/sec}$$

$$n = 0.5$$

$$k = 0.001 \text{ m/sec}$$

**Profile C**

5343648



$$n = 0.23$$

$$k = 0.000009 \text{ m/sec}$$

Fig. 3 Soil profiles and properties

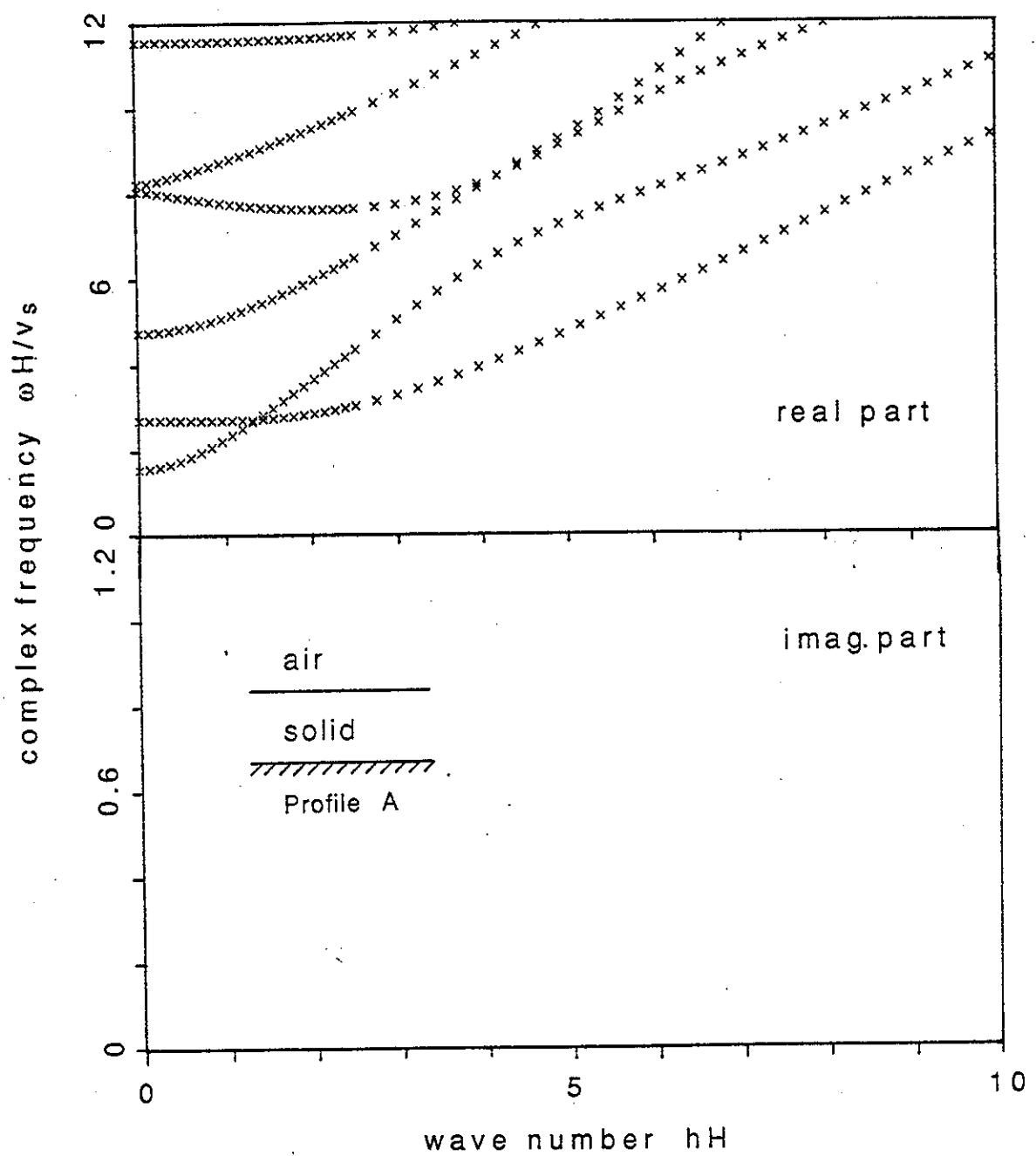
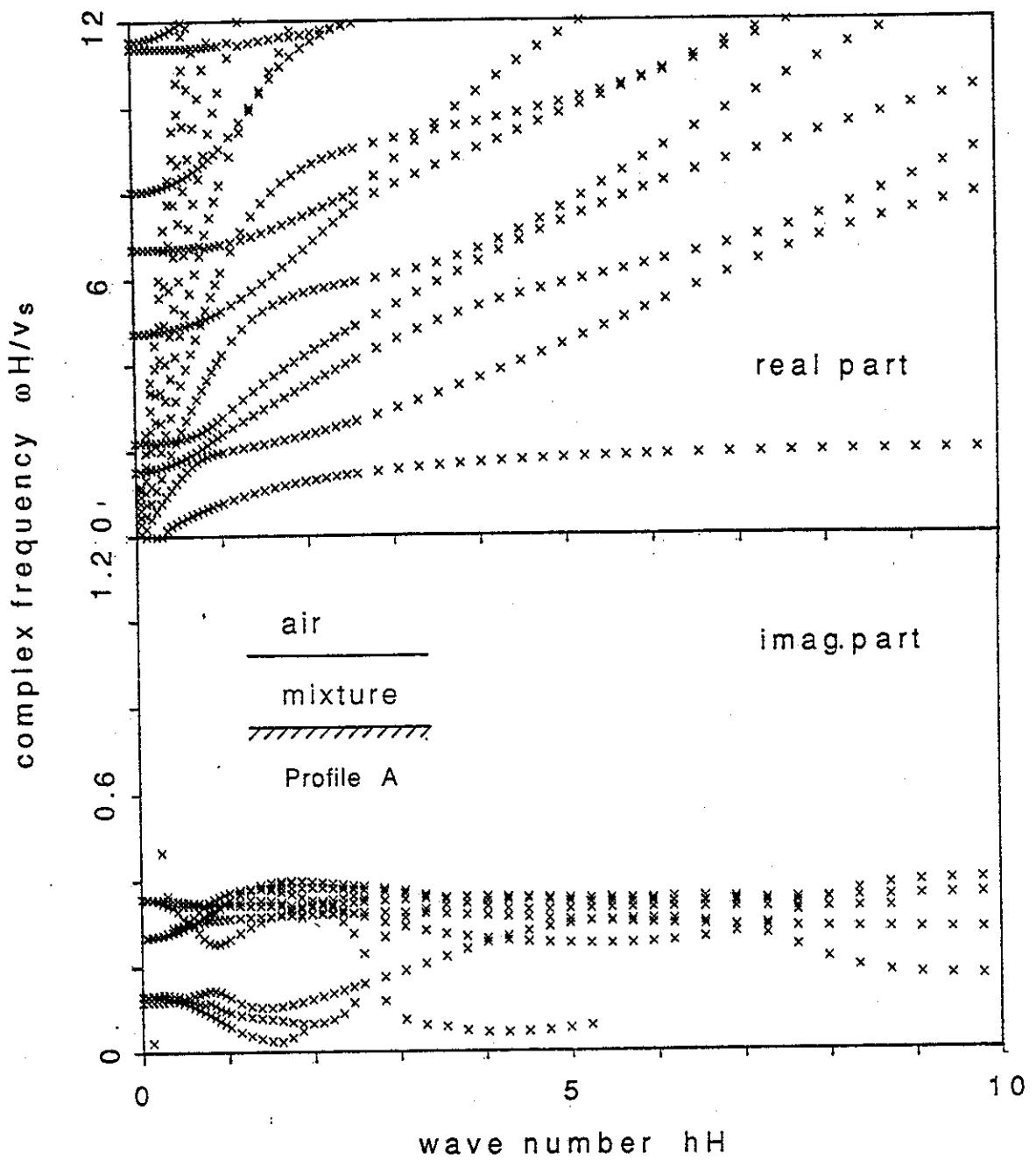
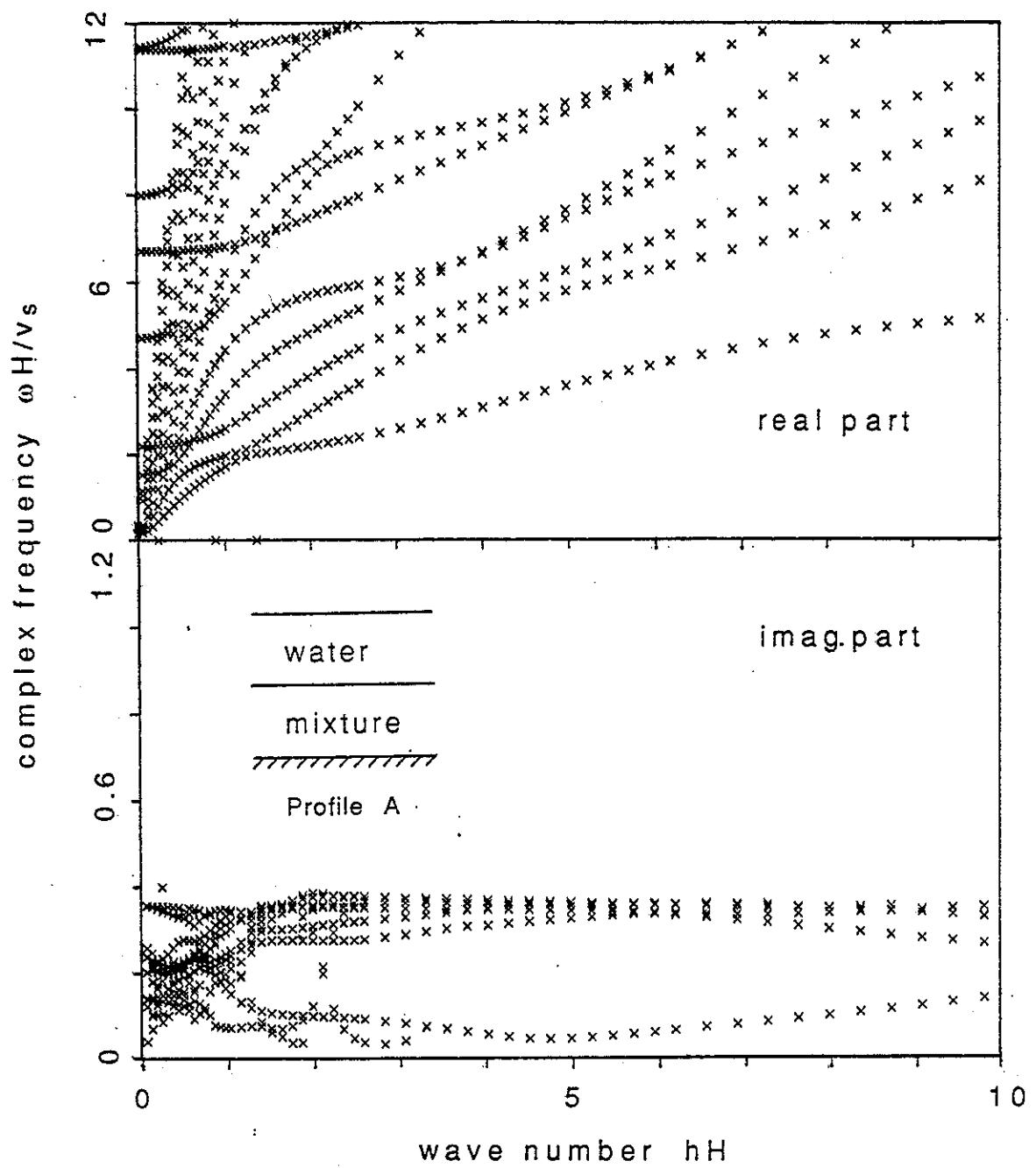


Fig. 4a Rayleigh wave dispersion curves, Profile A  
(a) air-solid



(b) air-mixture



(c) water-mixture

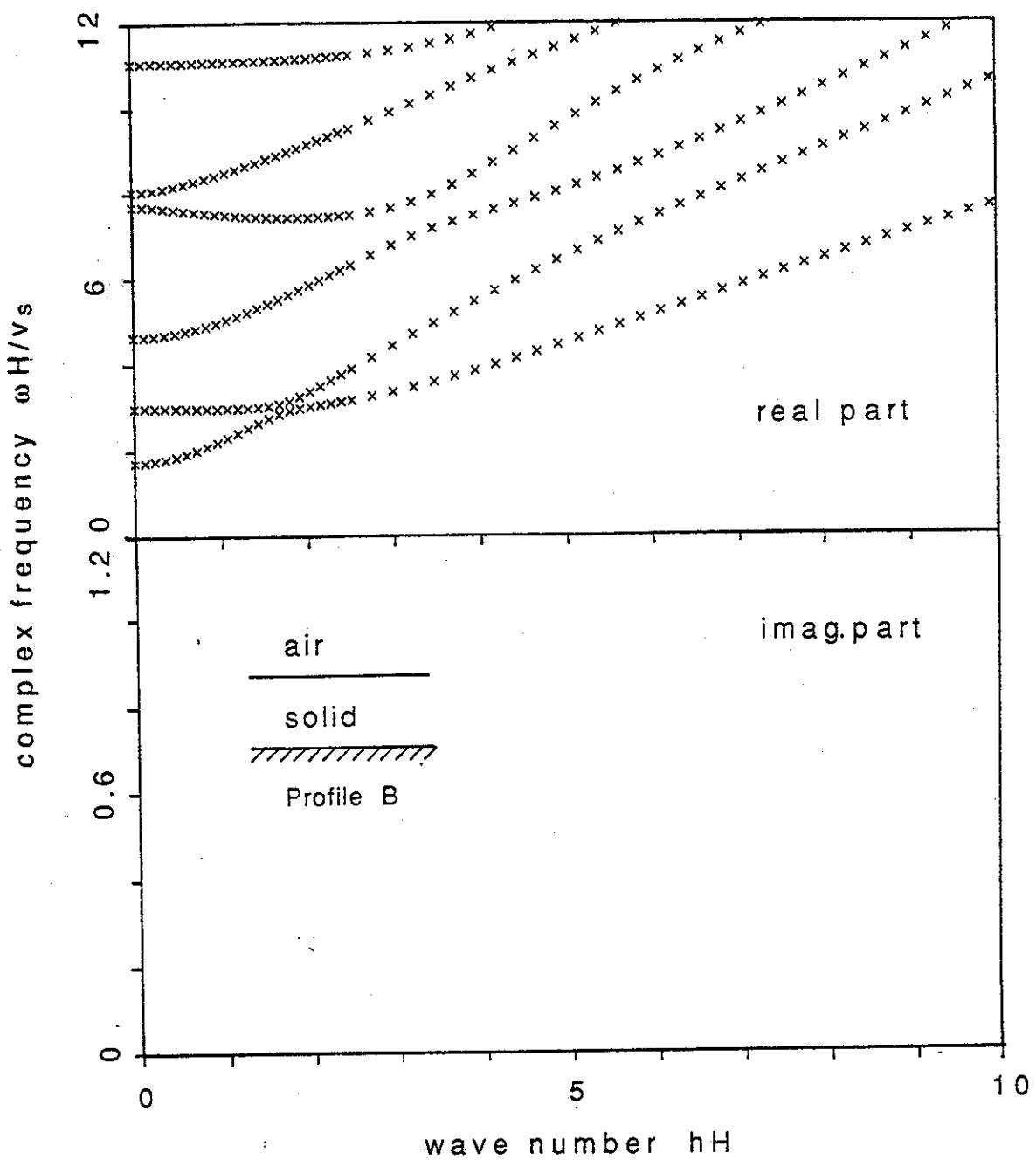
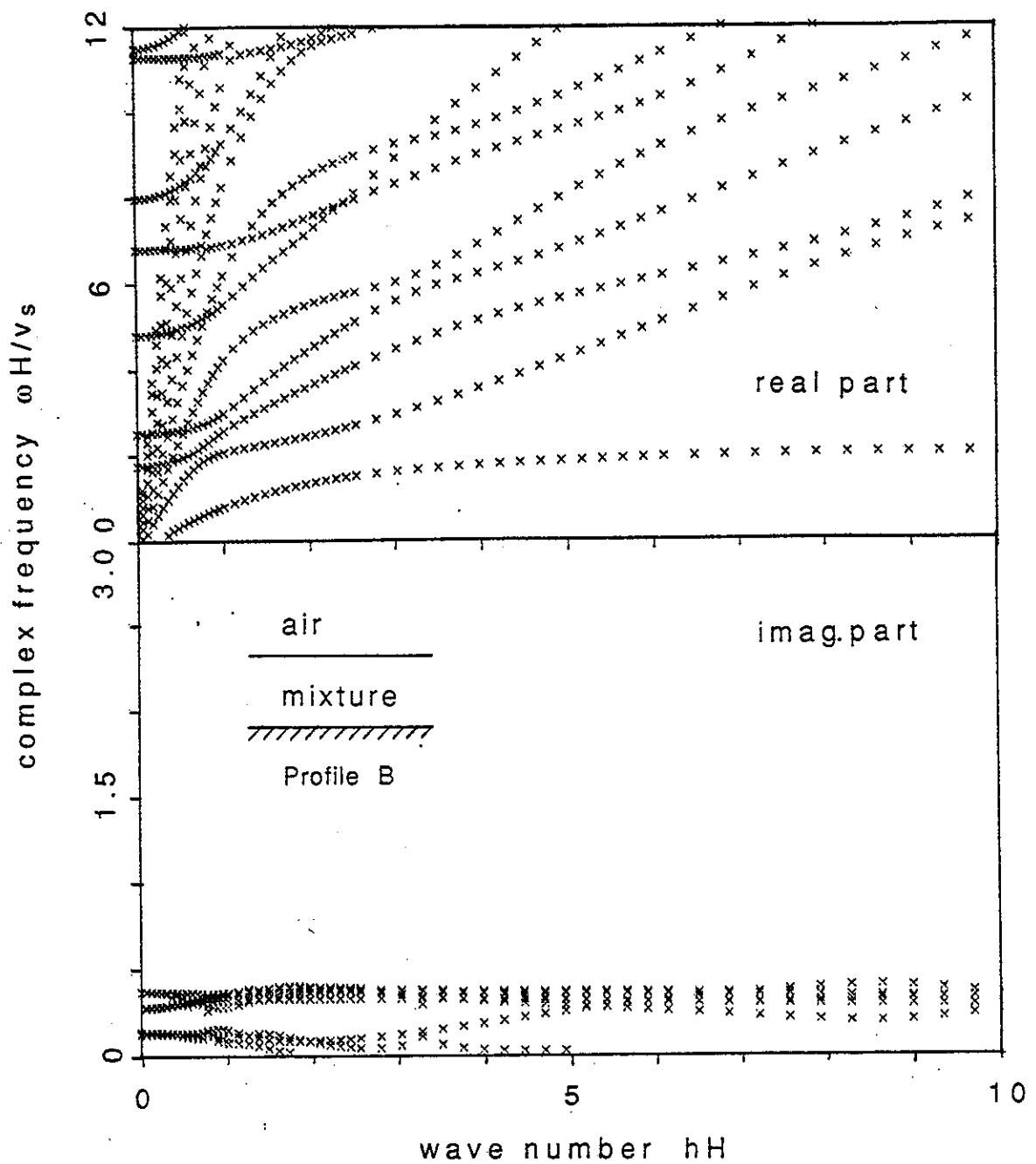
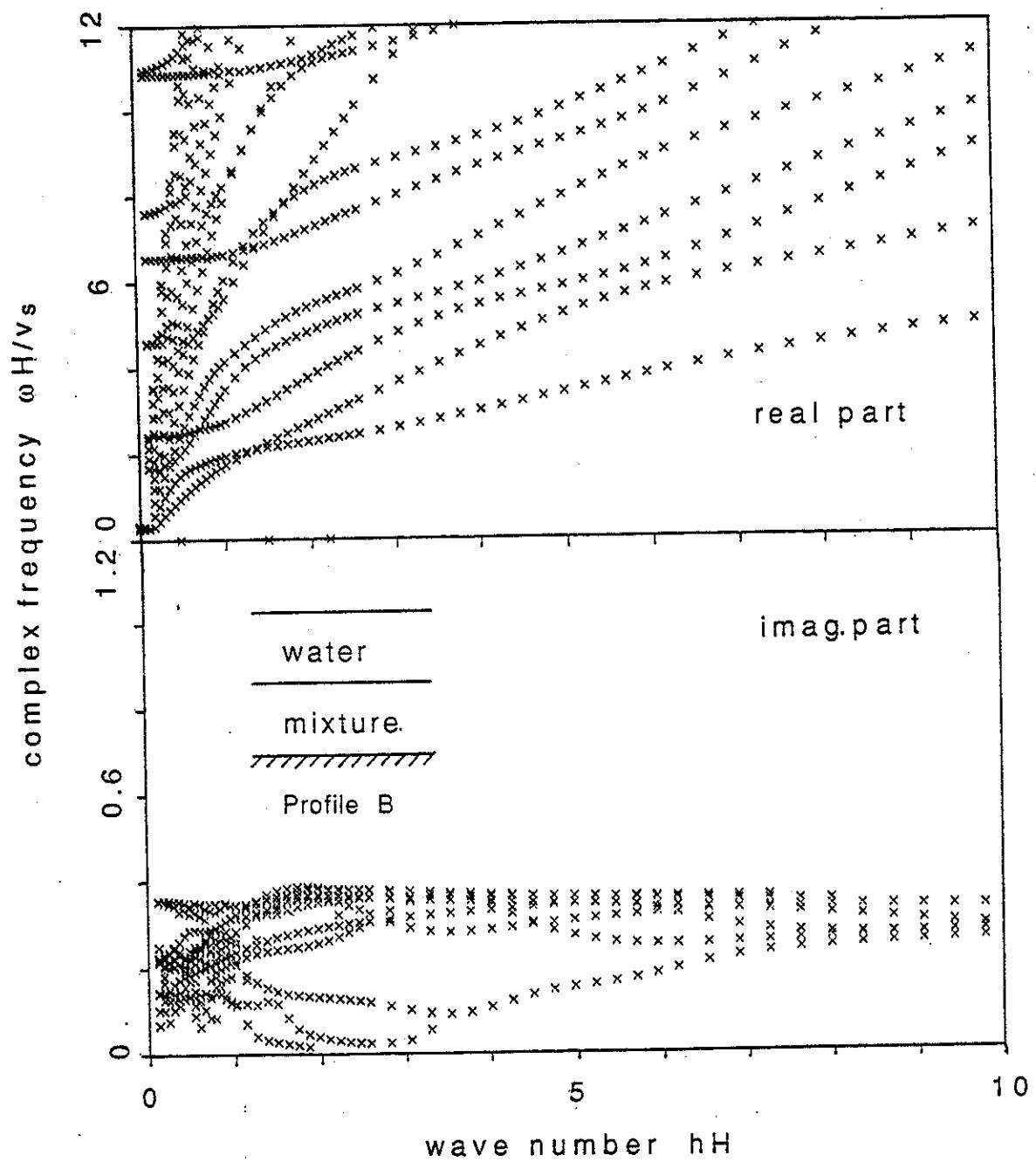


Fig. 4b Rayleigh wave dispersion curves, Profile B  
(a) air-solid



(b) air-mixture



(c) water-mixture

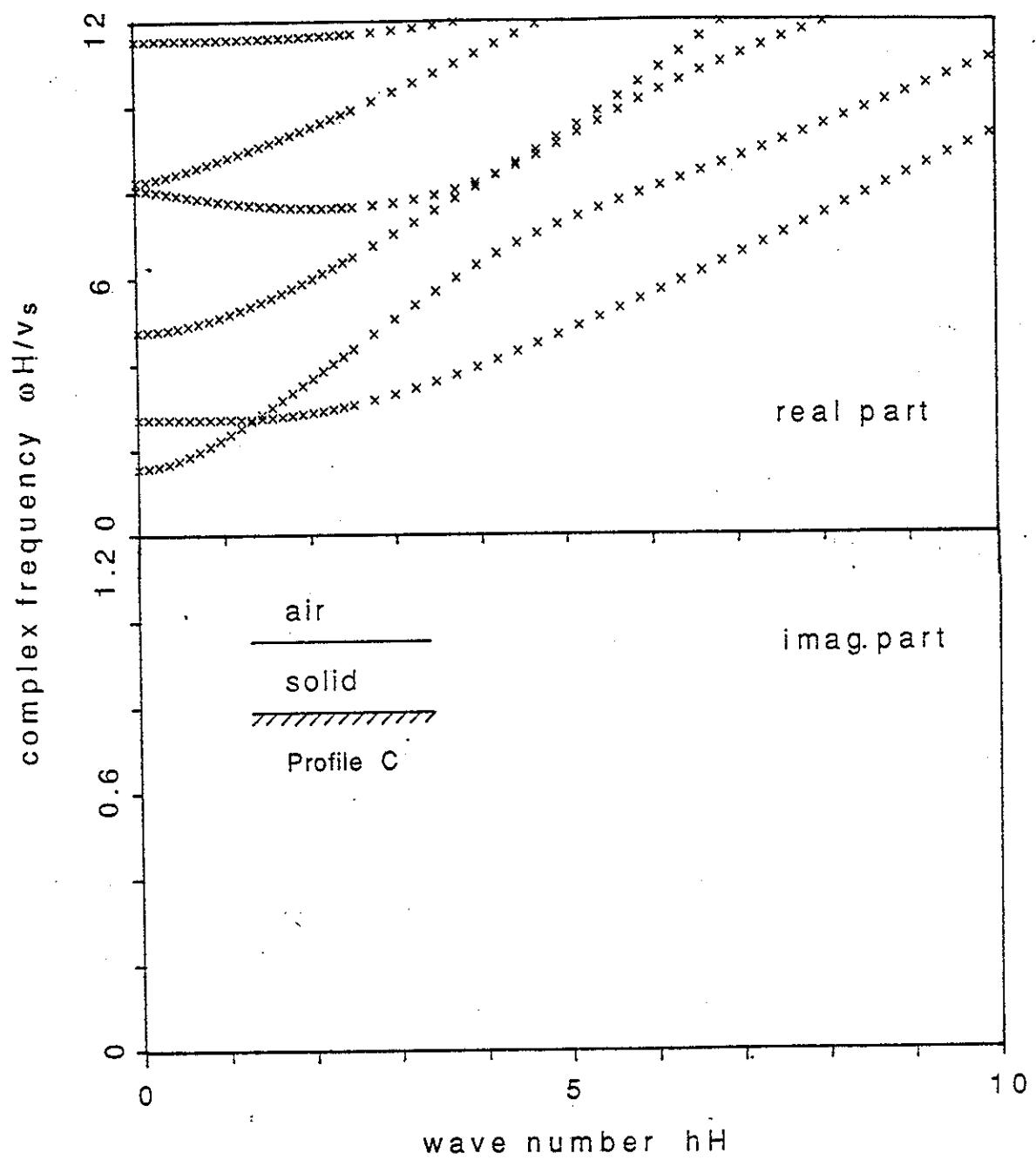
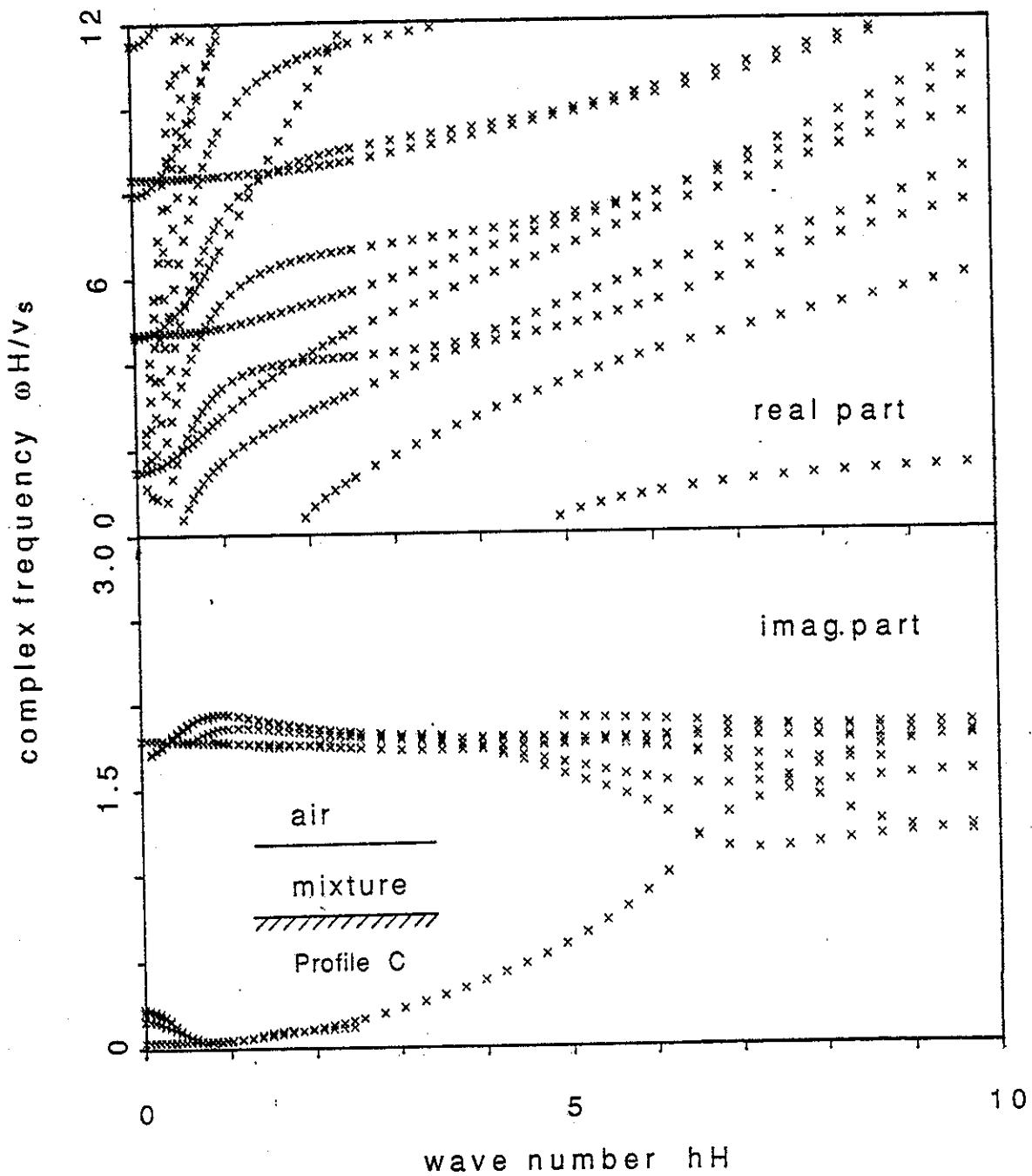


Fig. 4c Rayleigh wave dispersion curves, Profile C  
(a) air-solid



(b) air-mixture

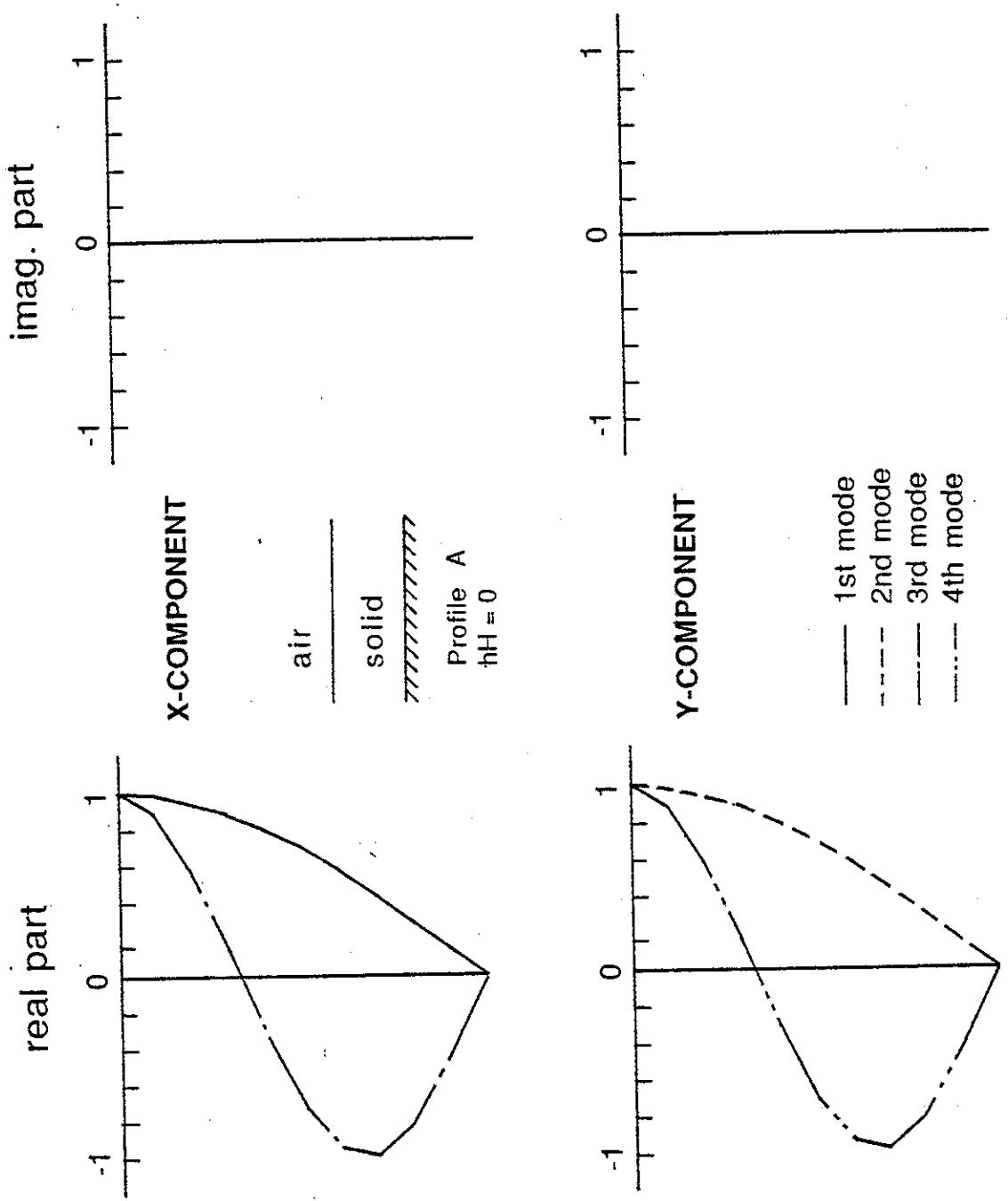
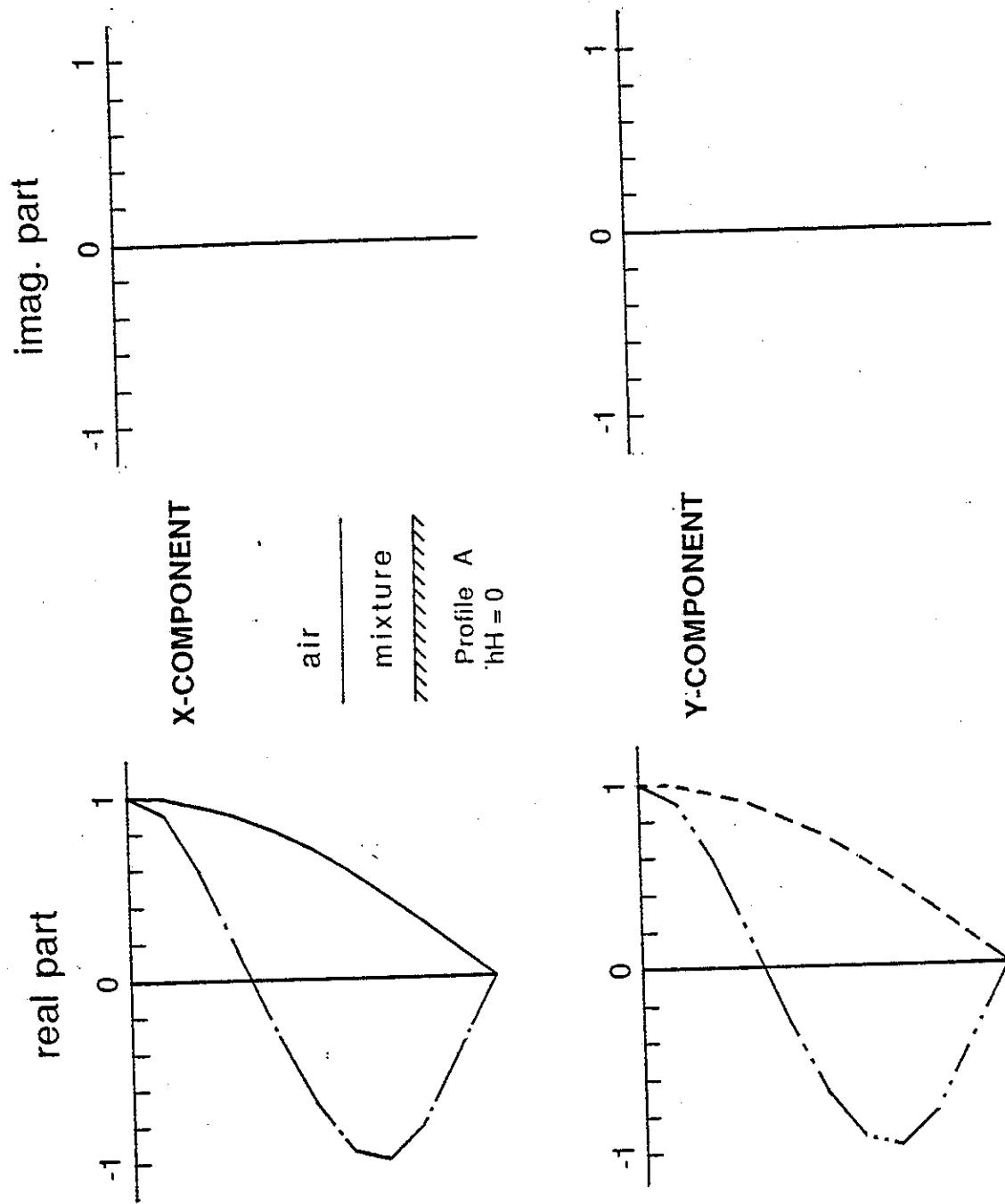
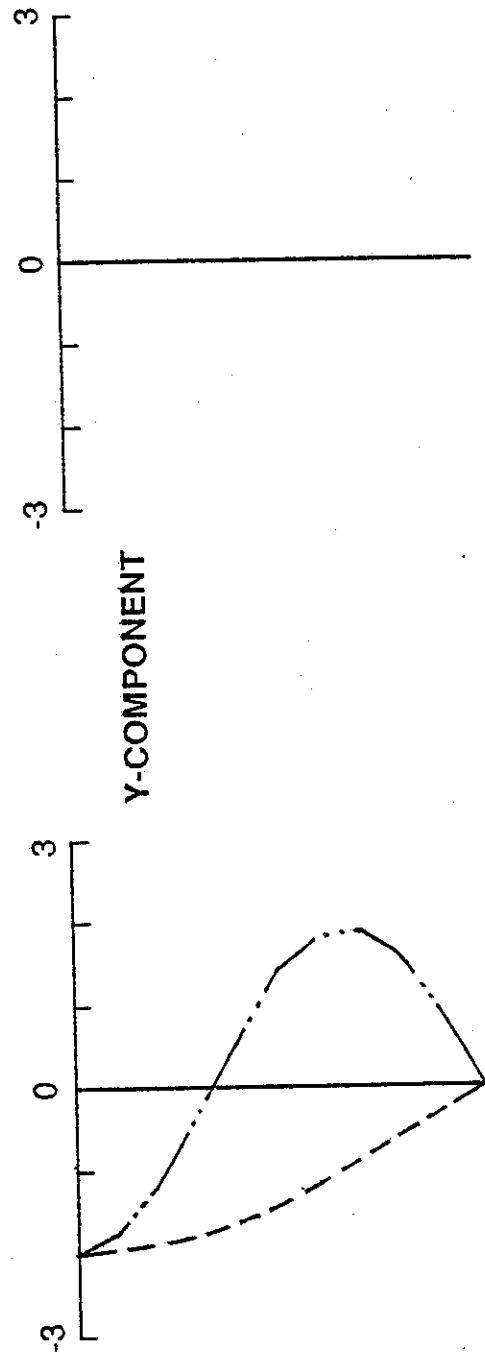
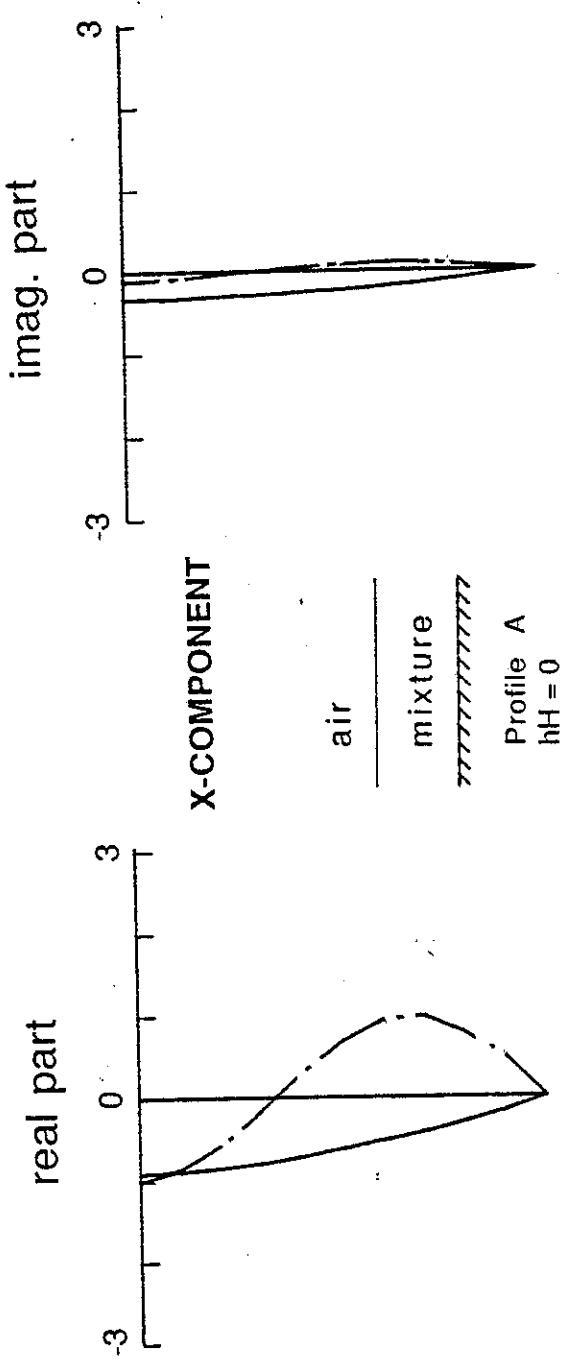


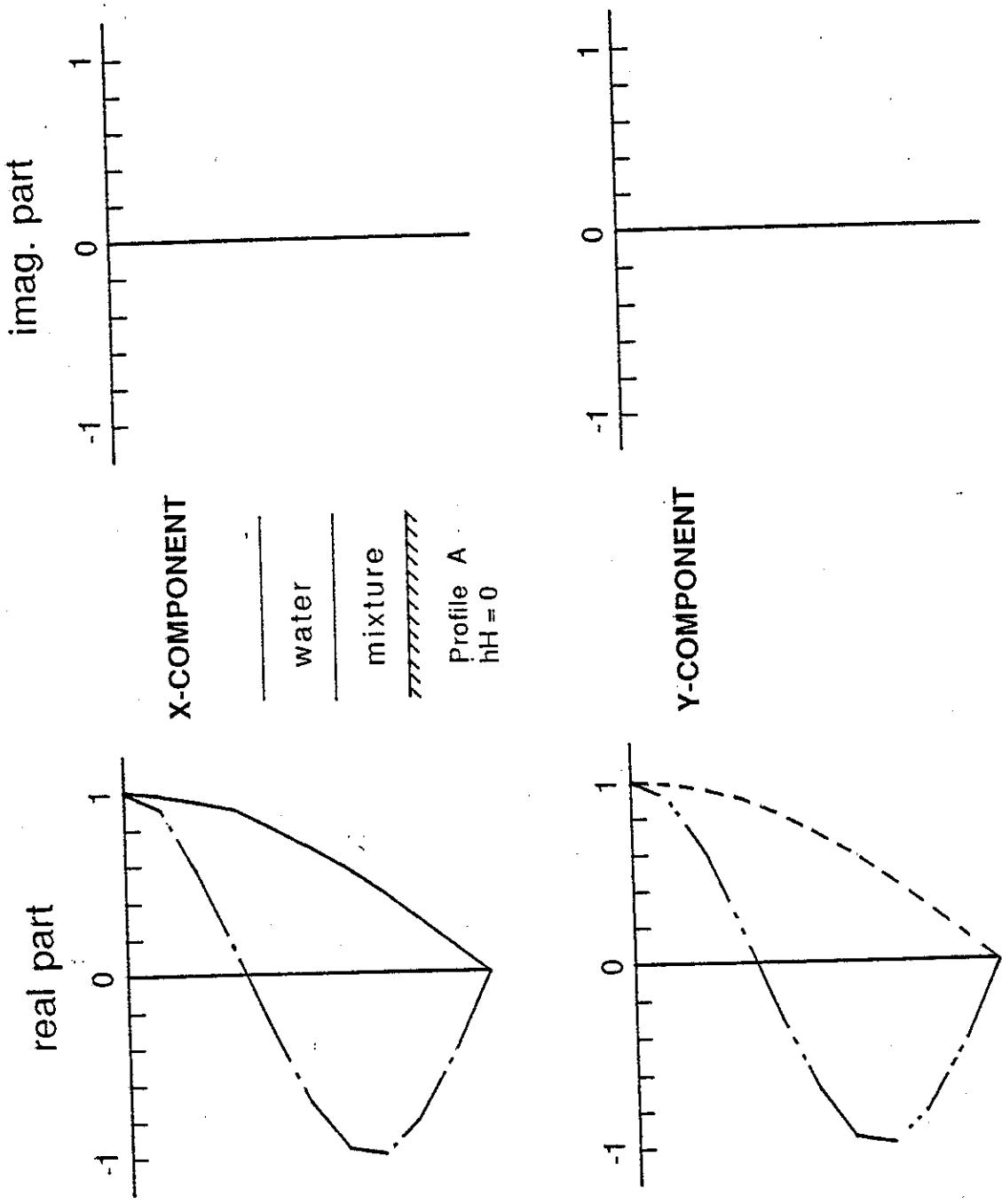
Fig. 5a Normalized displacement distributions for  $hH = 0$ , Profile A  
 (a) air-solid

(b) air-mixture, solid frame

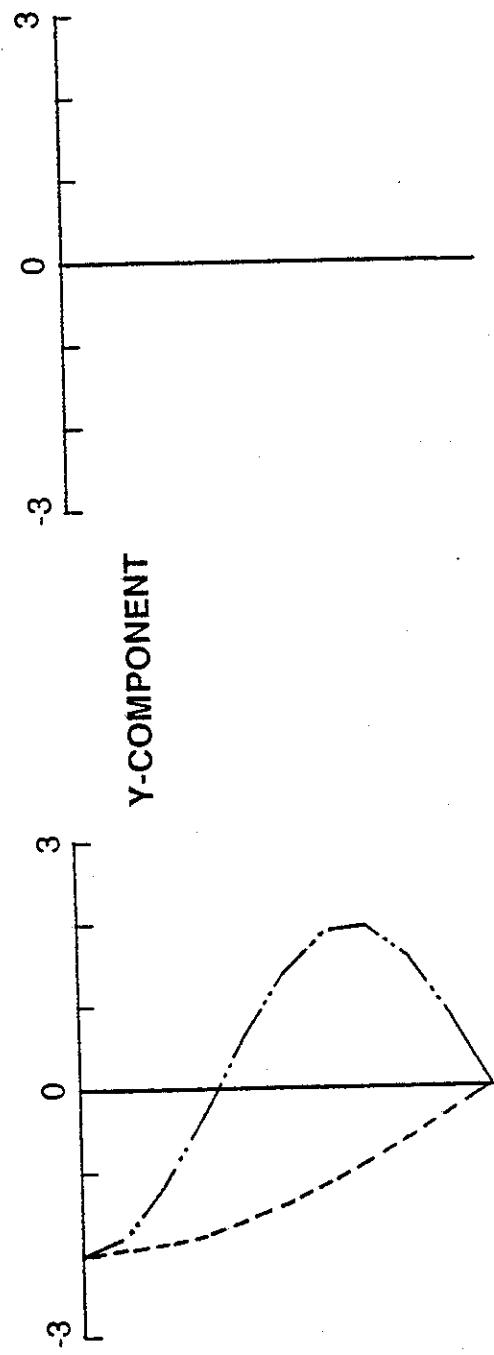
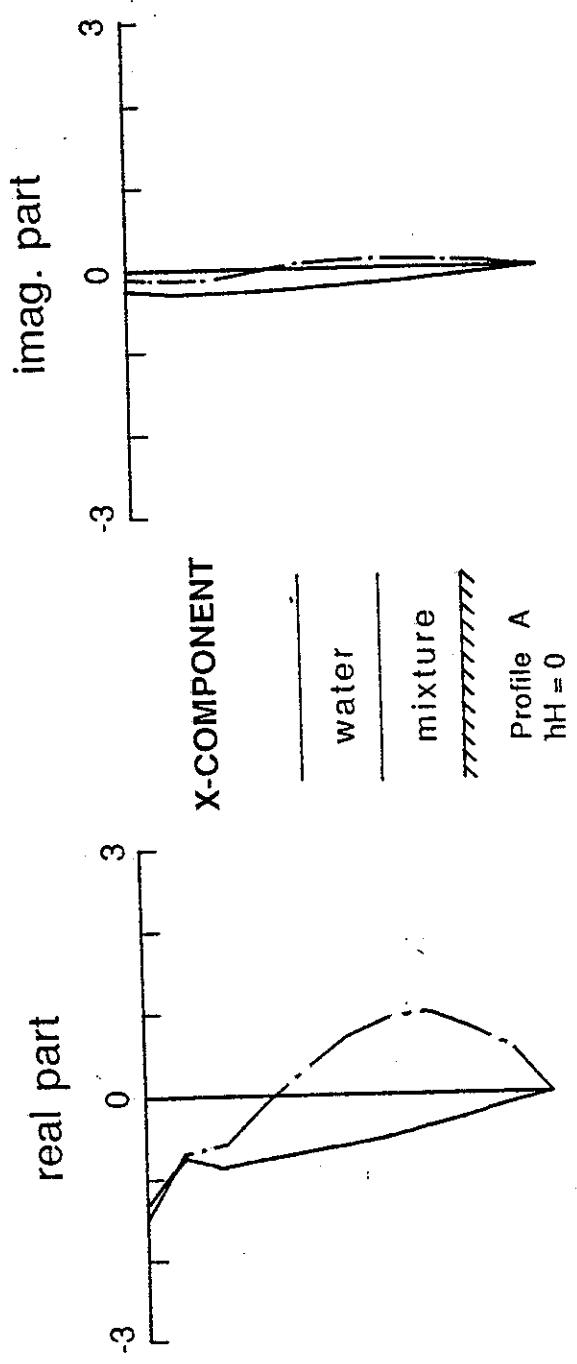




(c) air-mixture, fluid



(d) water-mixture, solid frame



(e) water-mixture, fluid

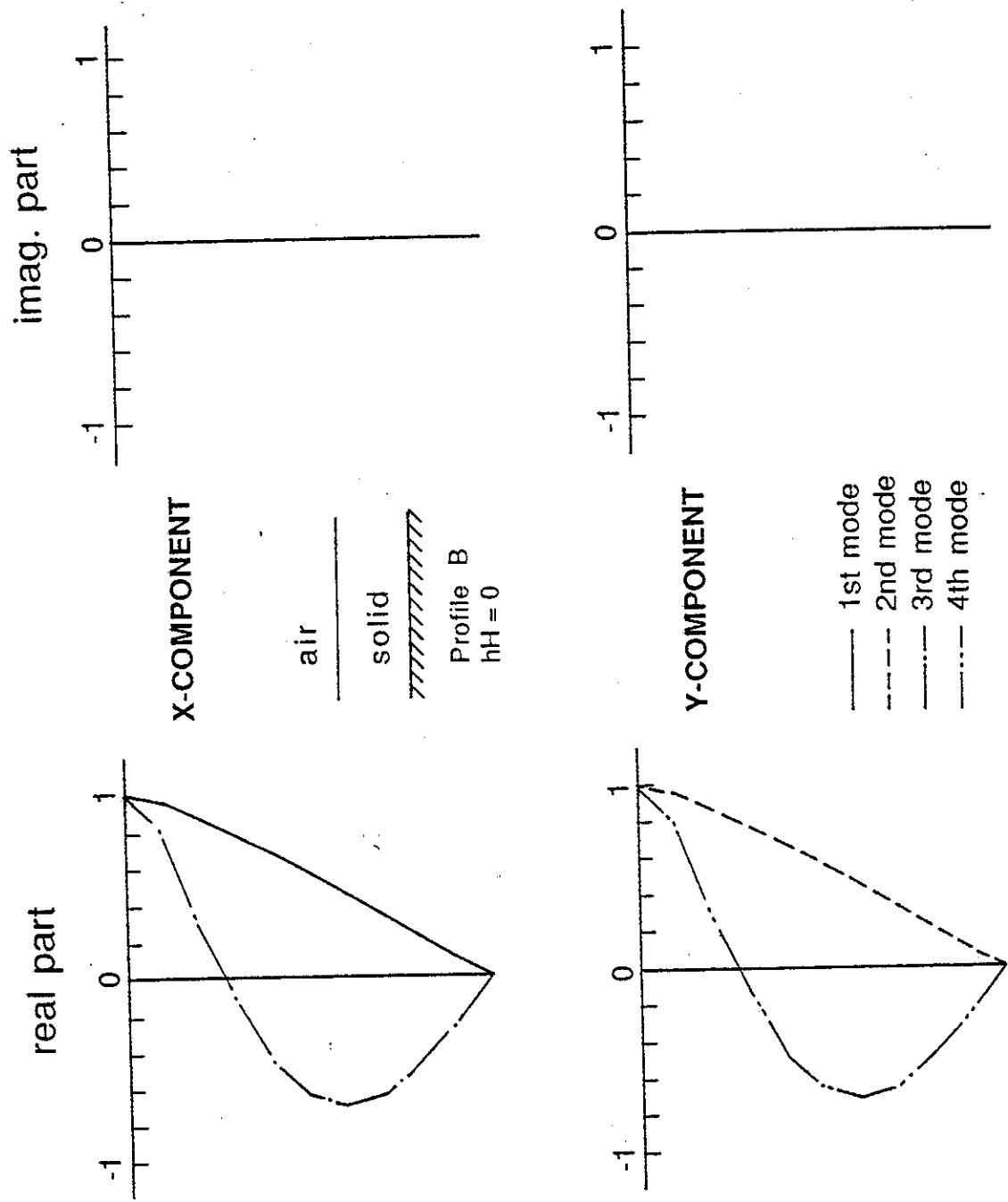
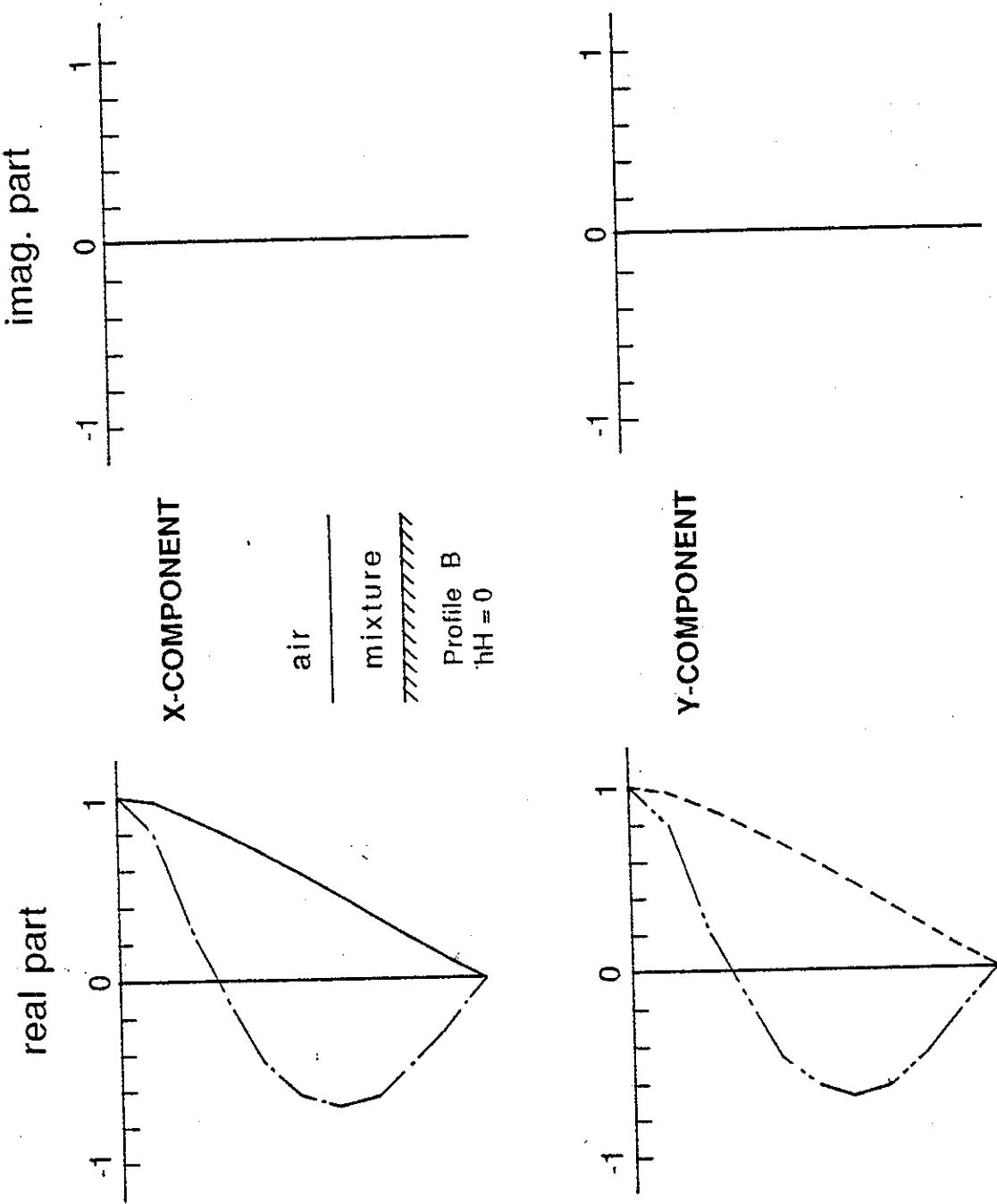
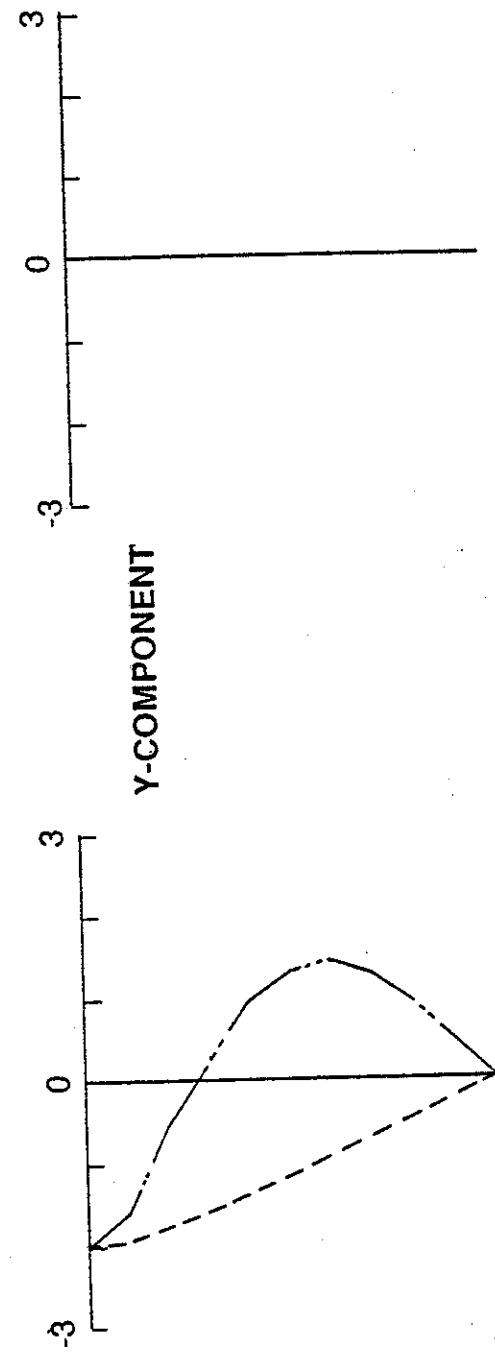
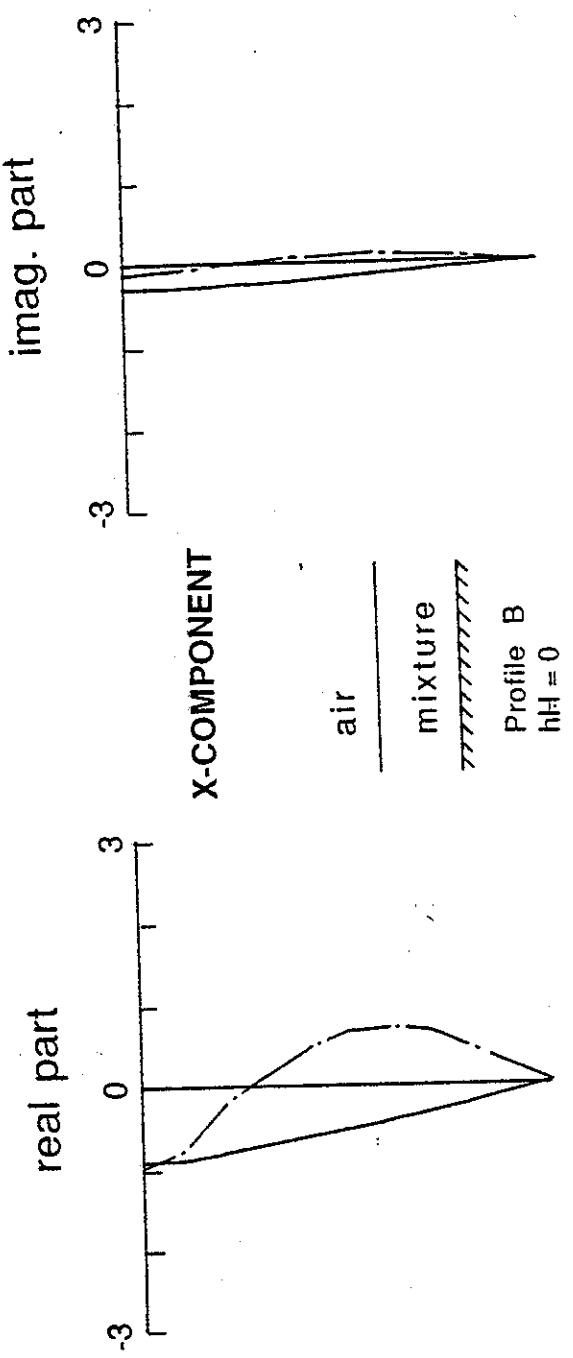


Fig. 5b Normalized displacement distributions for  $hH = 0$ , Profile B  
 (a) air-solid

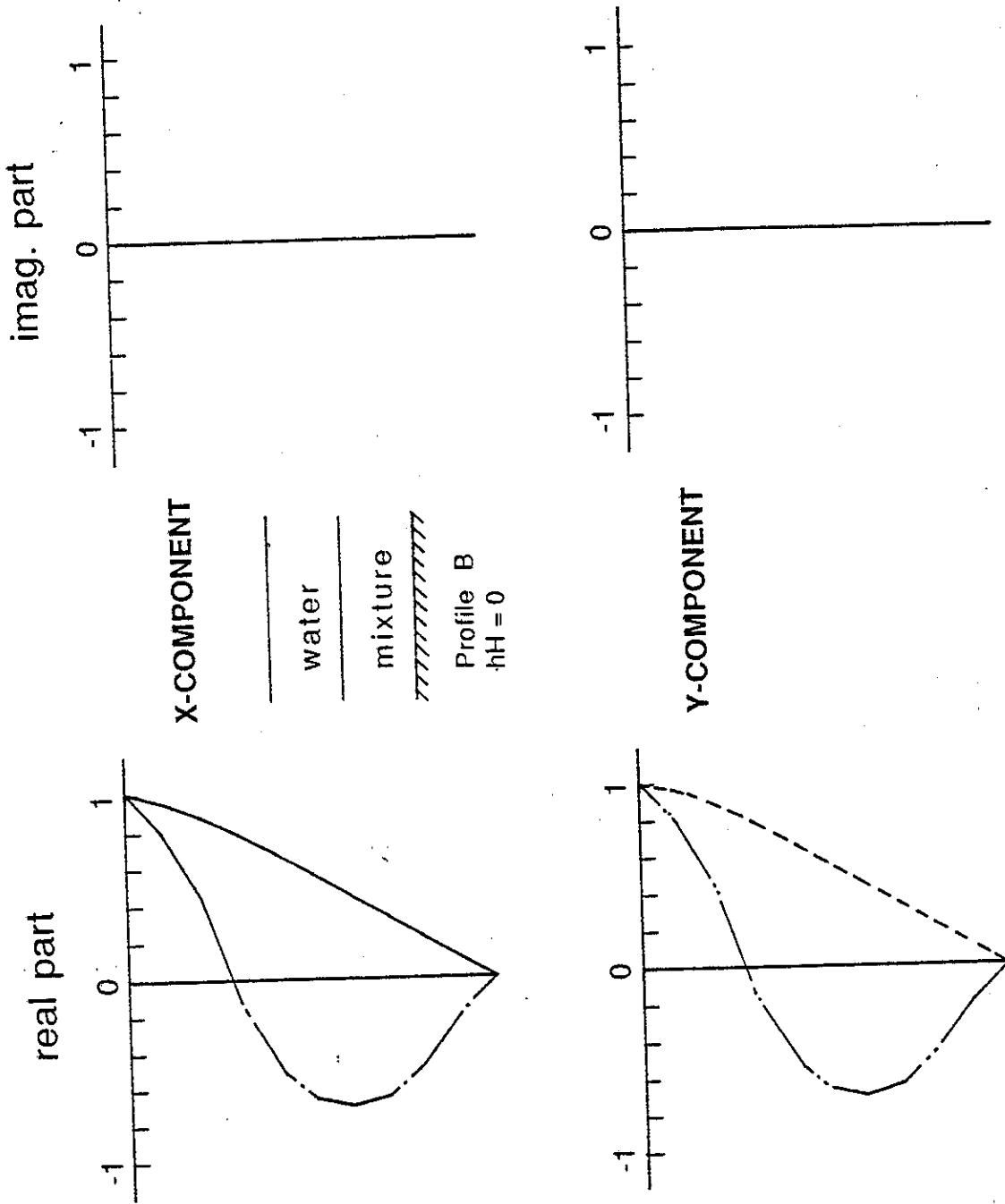


(b) air-mixture, solid frame

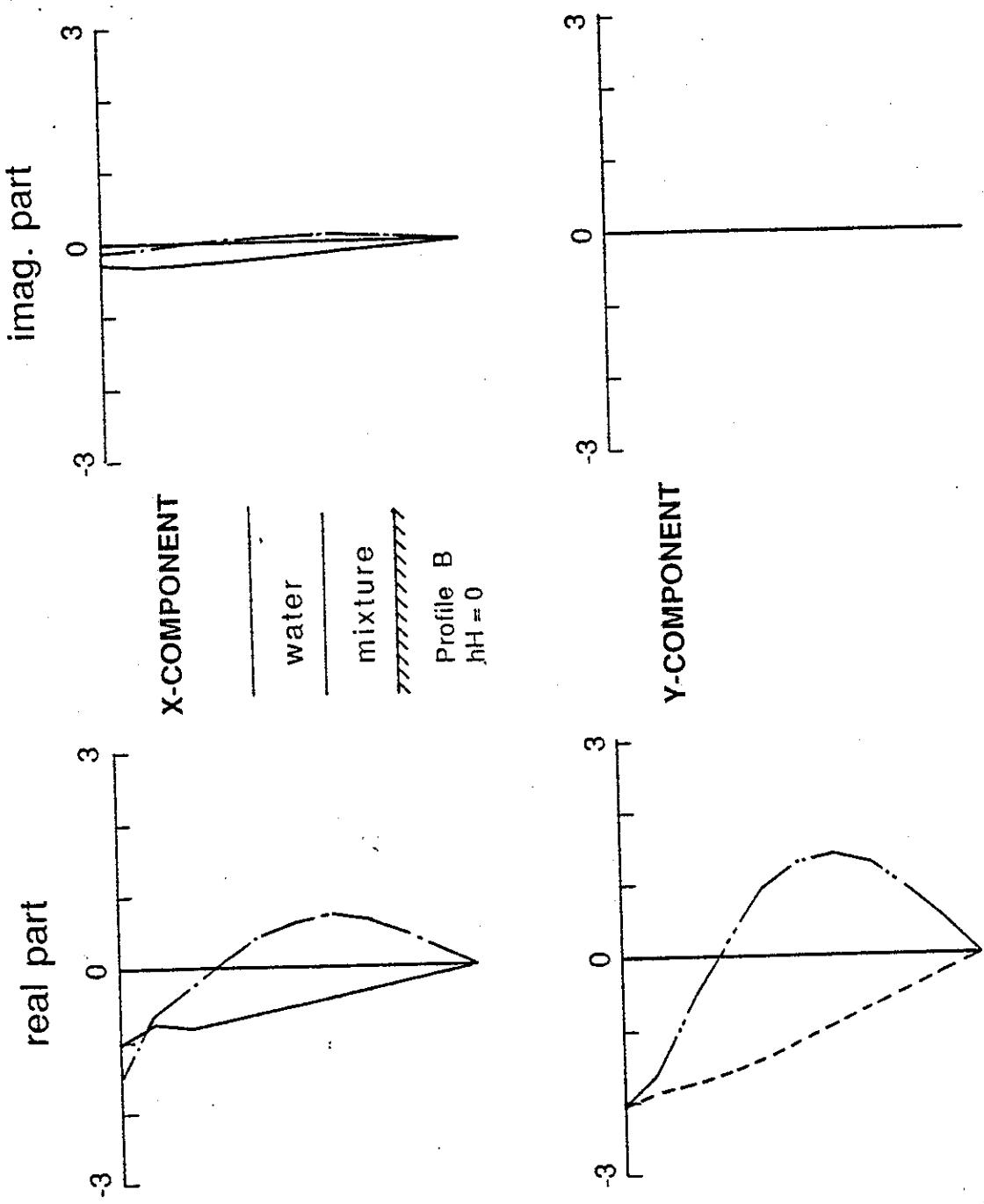


(c) air-mixture, fluid

(d) water-mixture, solid frame



(e) water-mixture, fluid



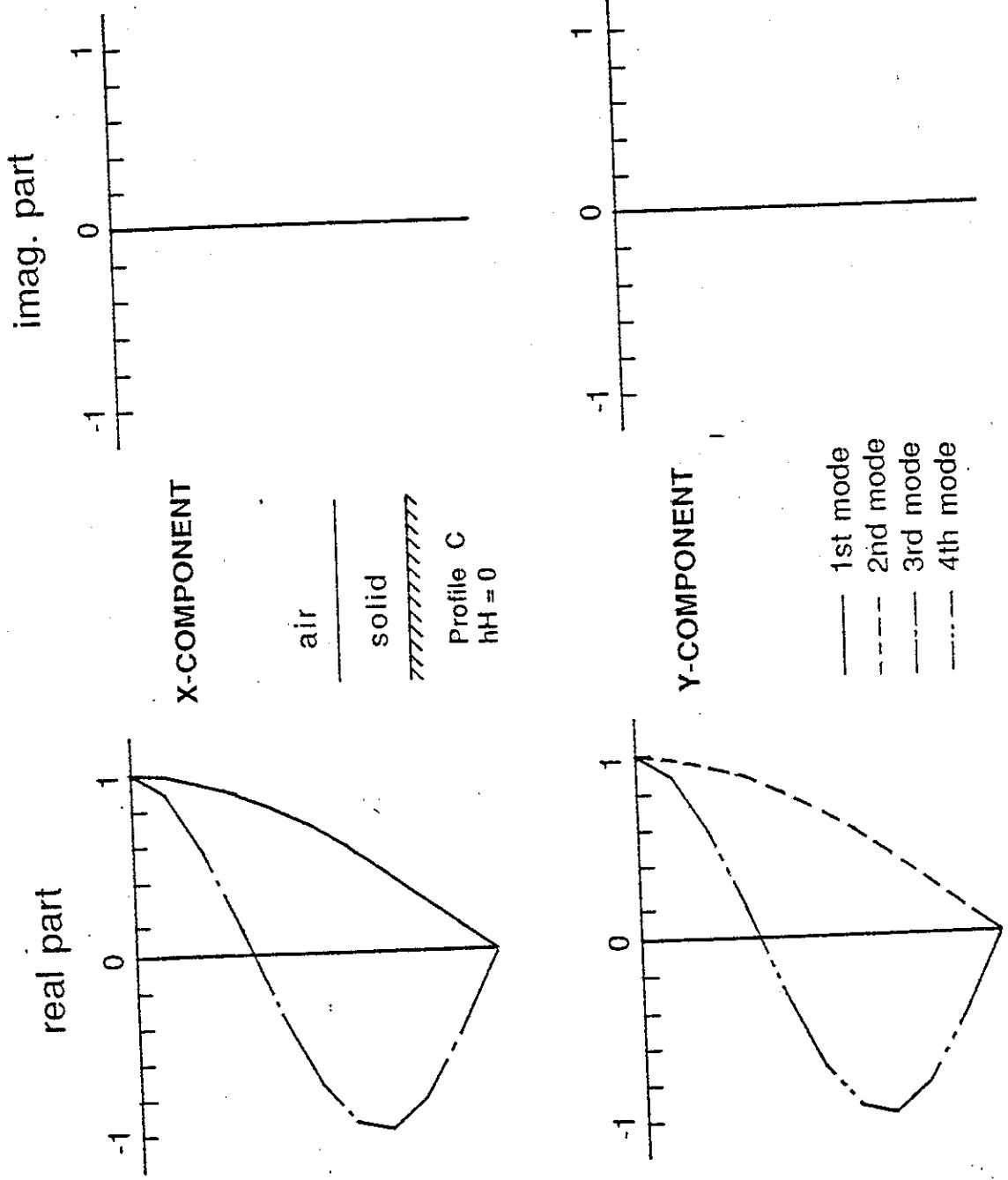
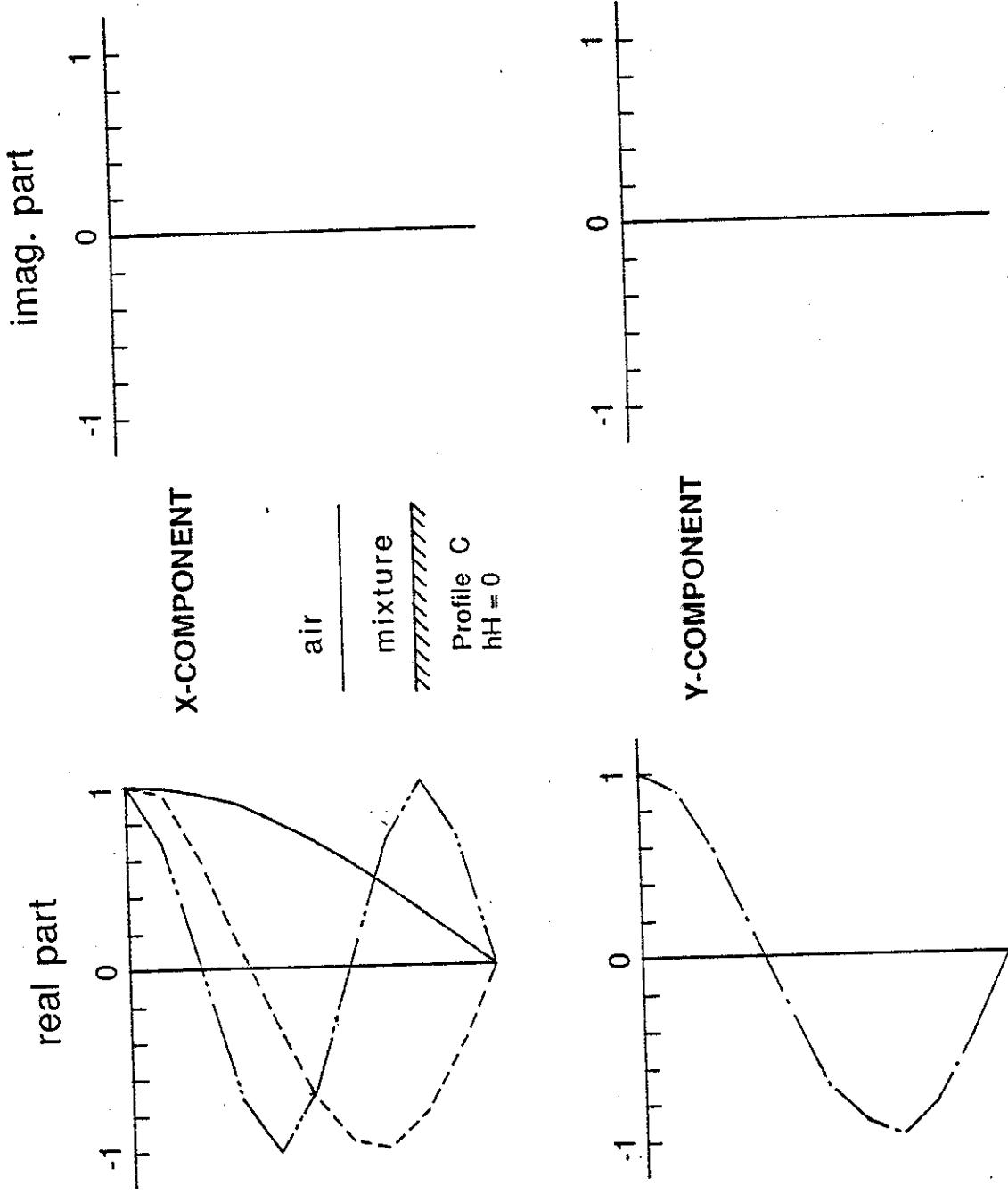
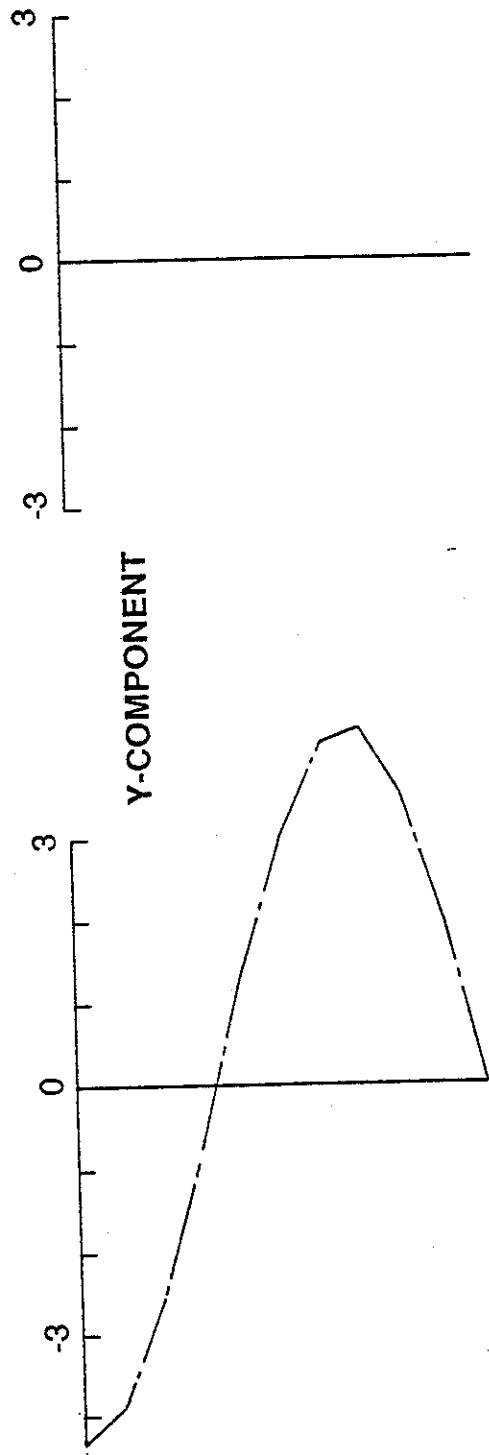
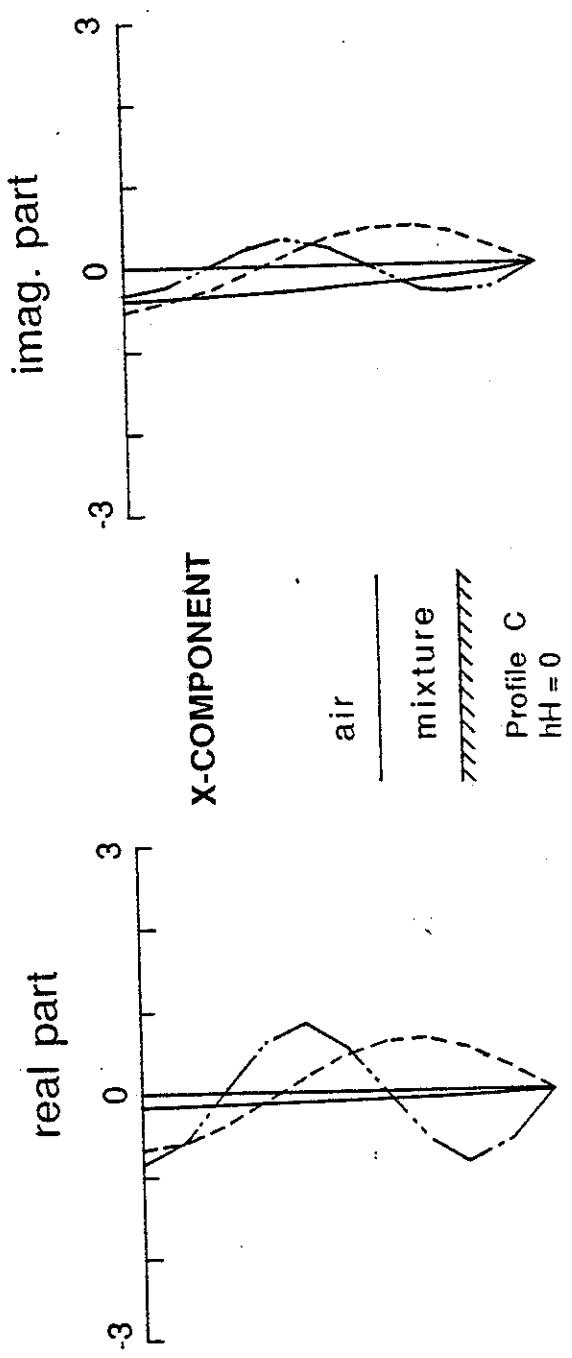


Fig. 5c Normalized displacement distributions for  $hH = 0$ , Profile C  
 (a) air-solid

(b) air-mixture, solid frame





(c) air-mixture, fluid

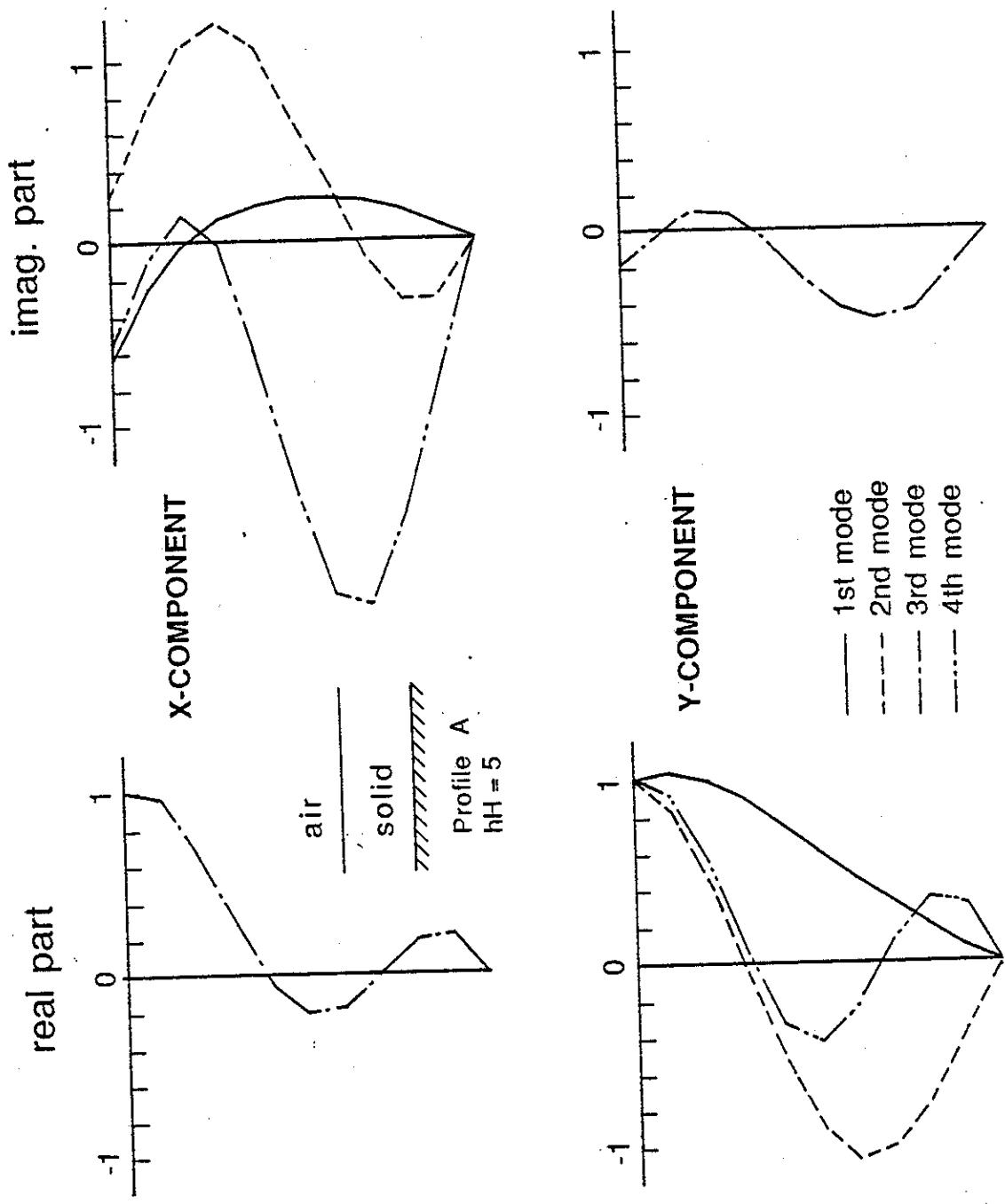
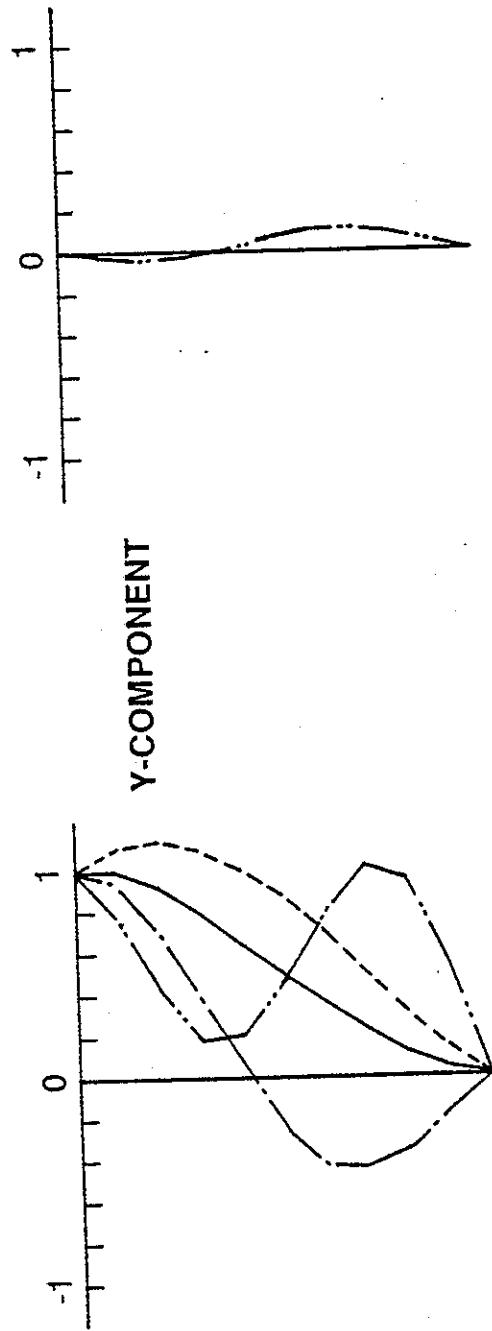
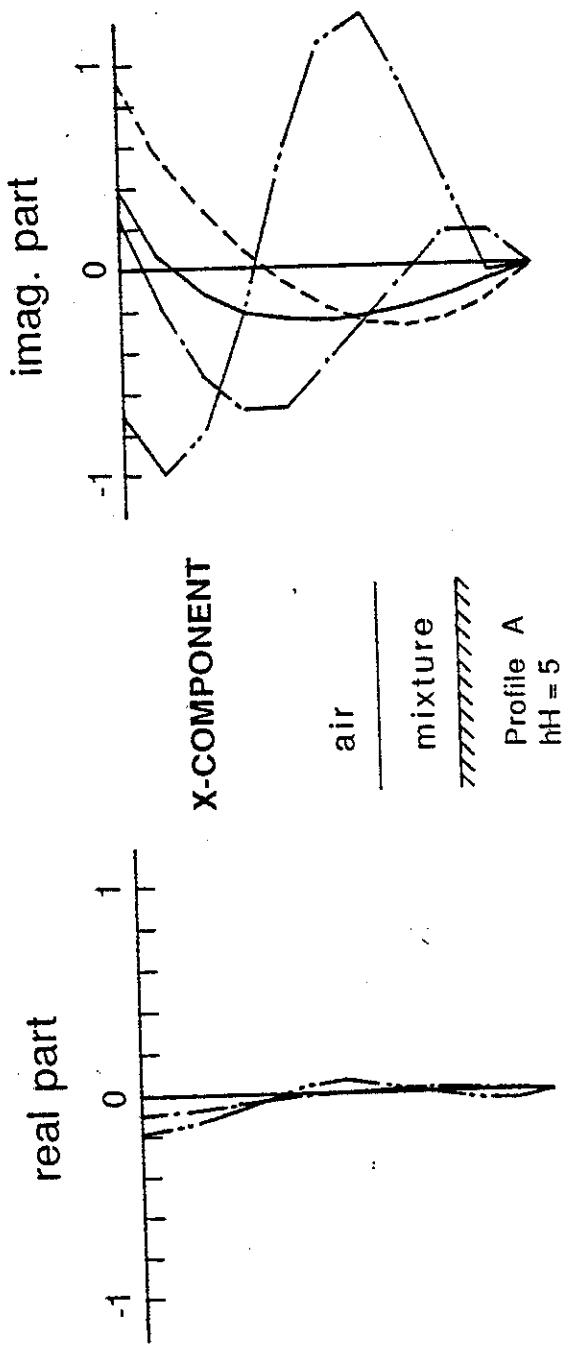
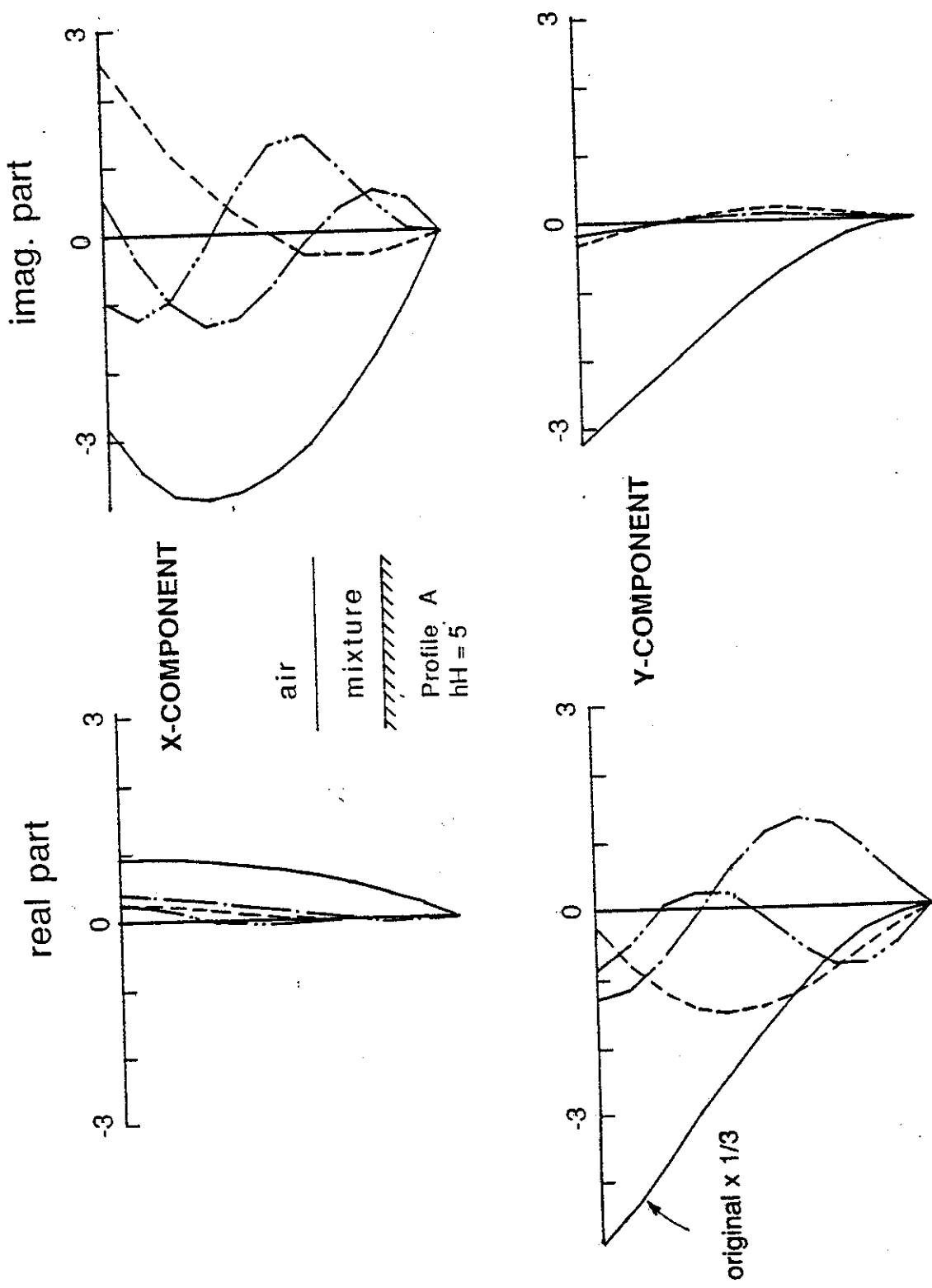


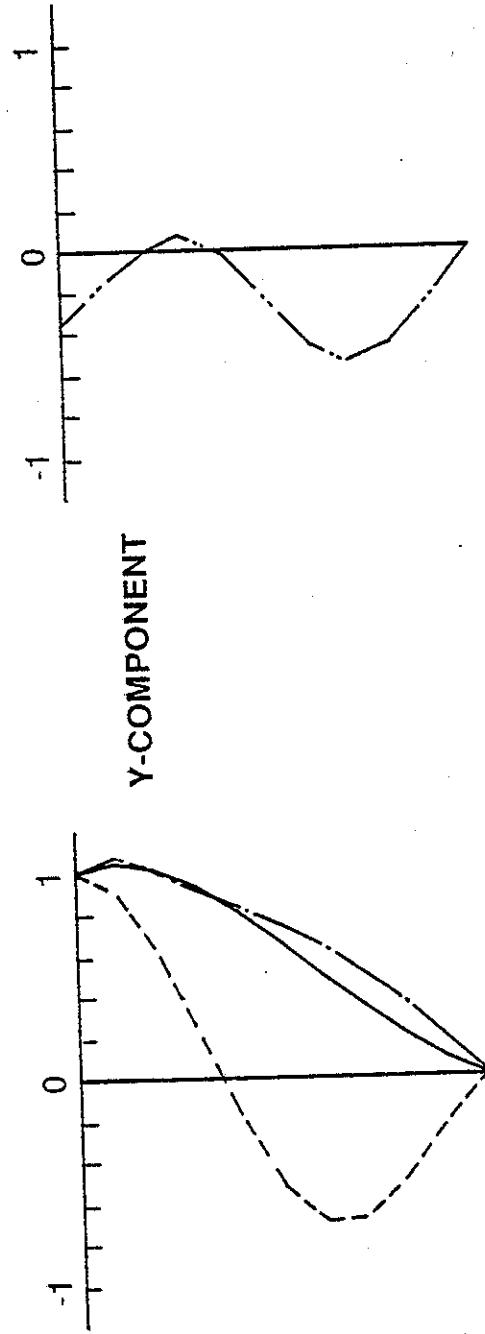
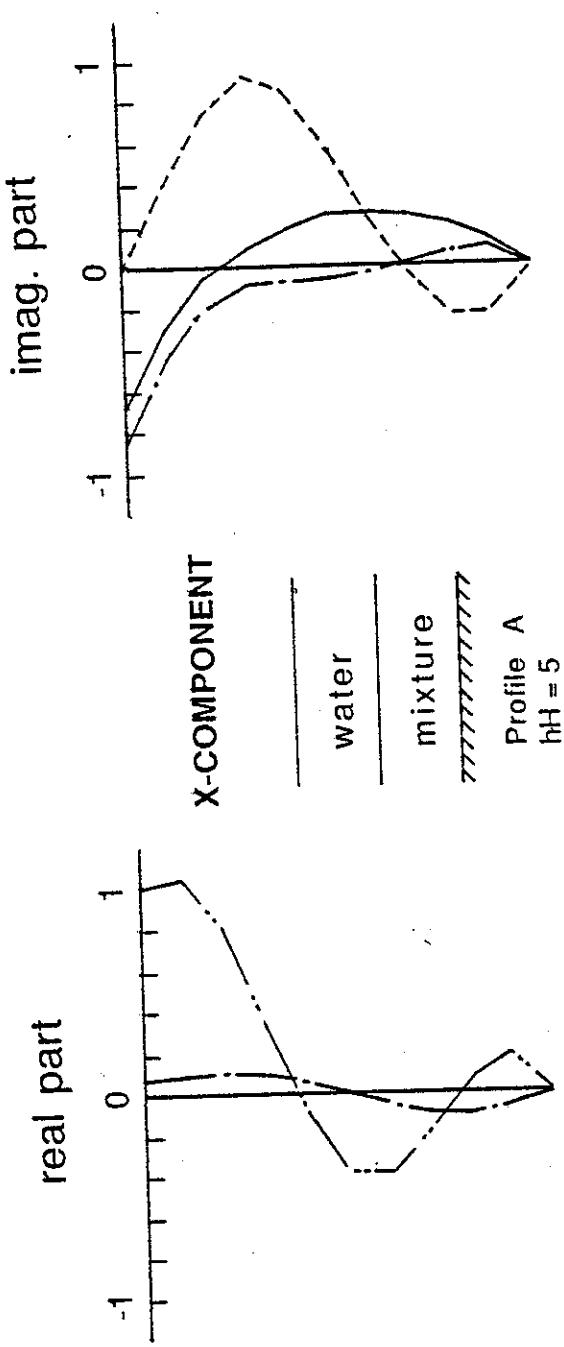
Fig. 6a Normalized displacement distributions for  $hH = 5$ , Profile A  
 (a) air-solid



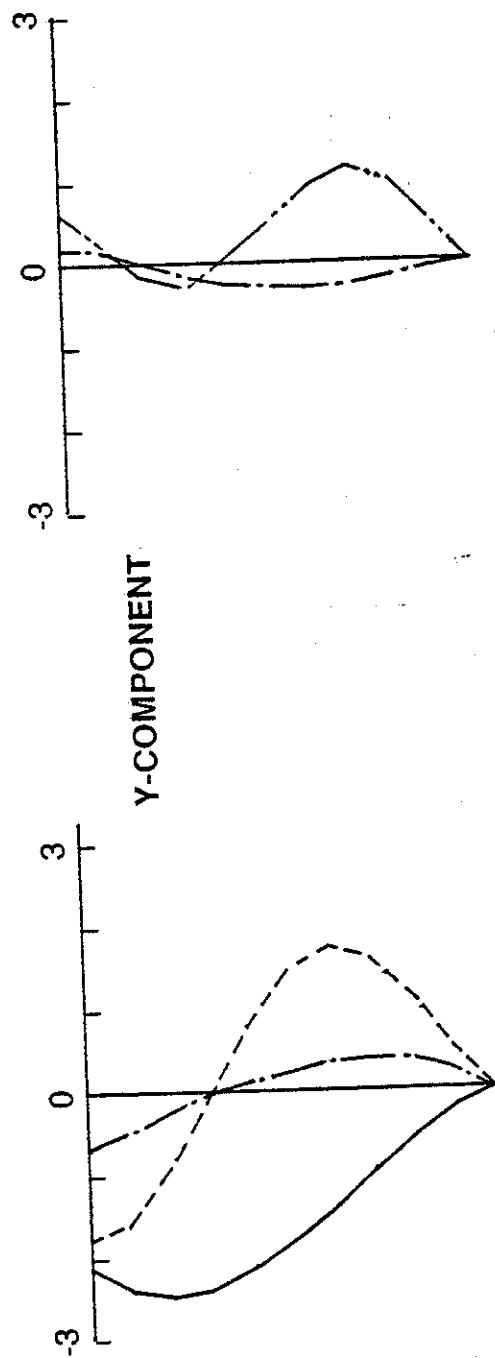
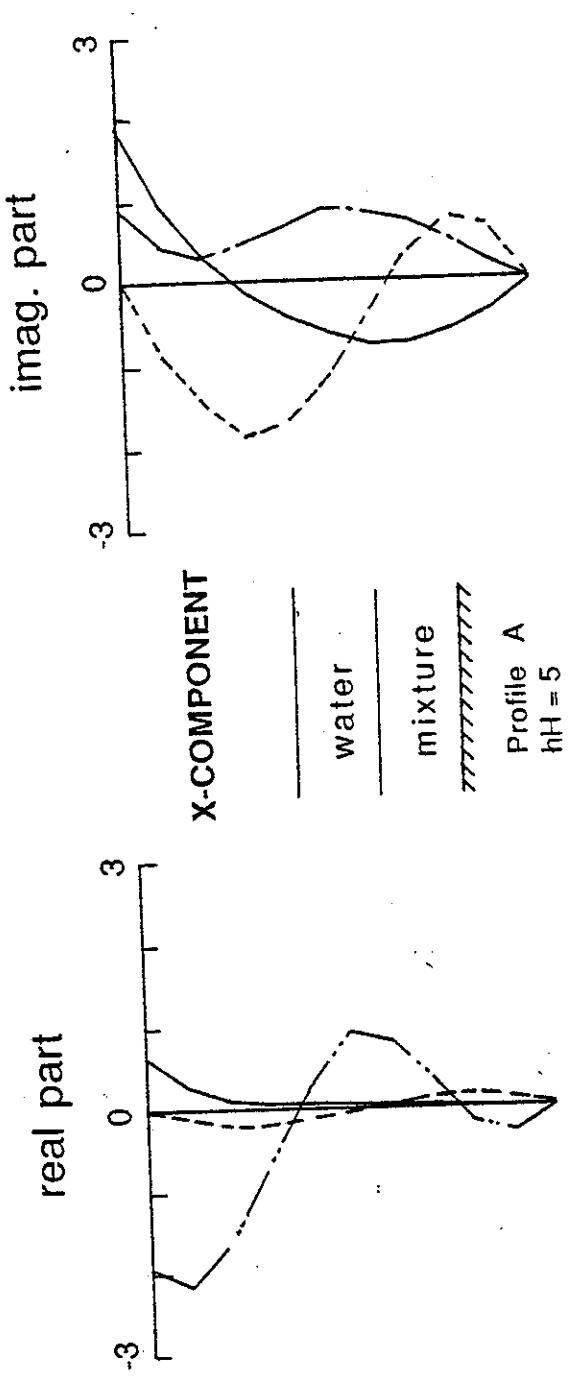
(b) air-mixture, solid frame



(c) air-mixture, fluid



(d) water-mixture, solid frame



(e) water-mixture, fluid

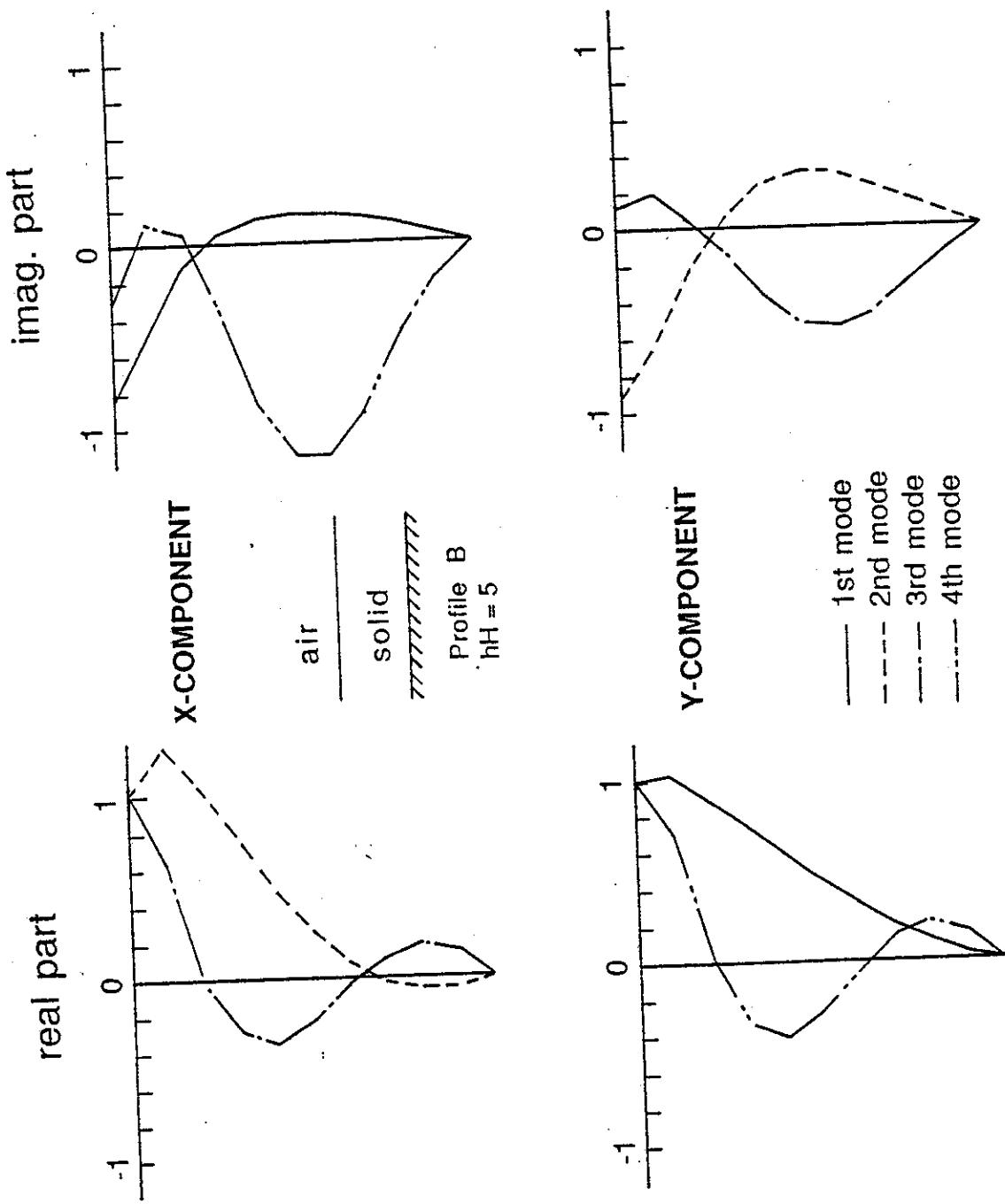
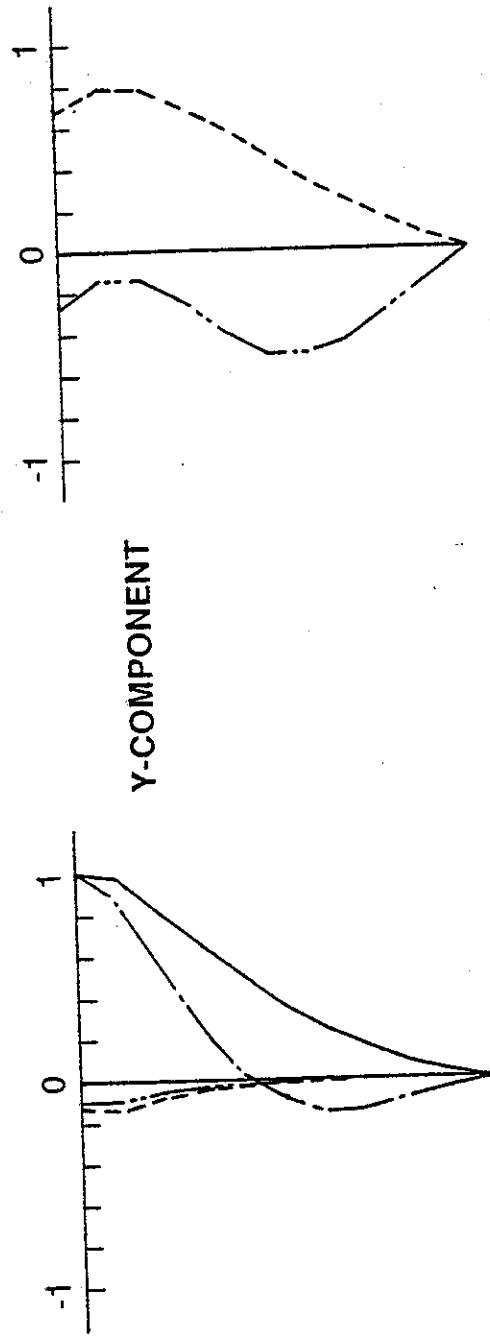
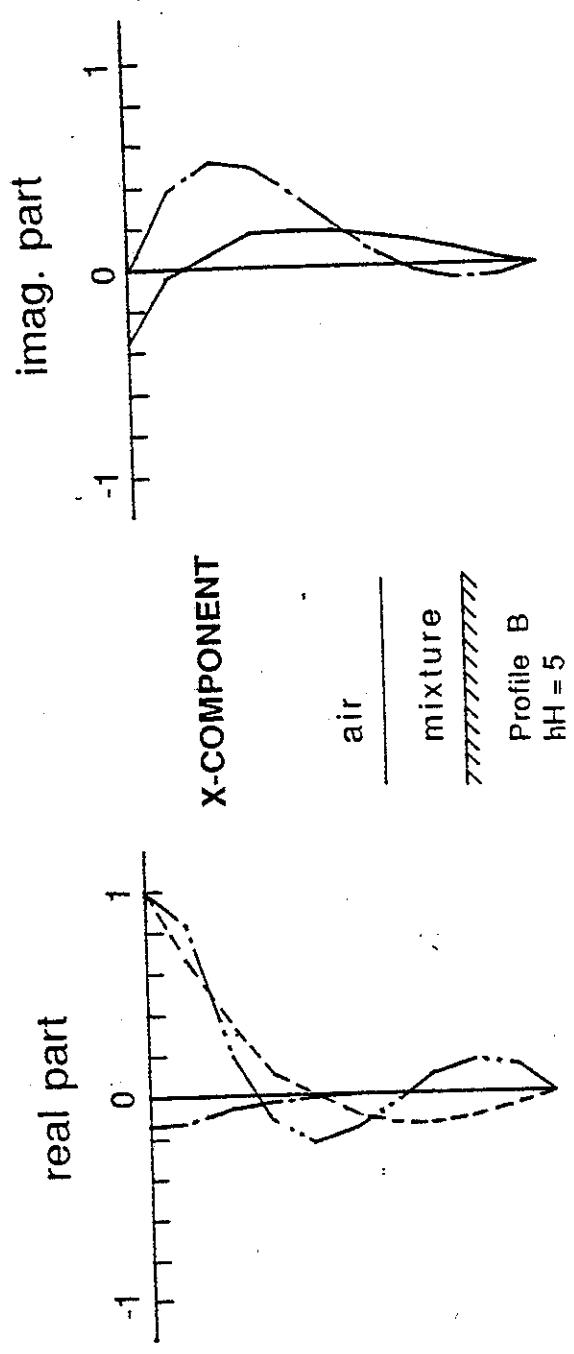
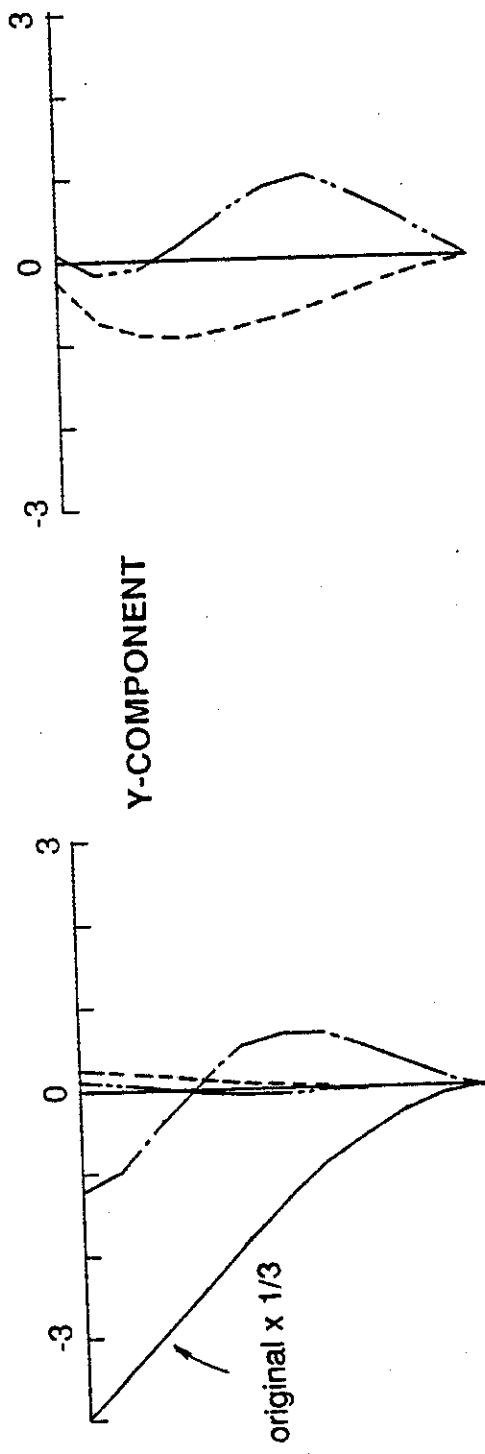
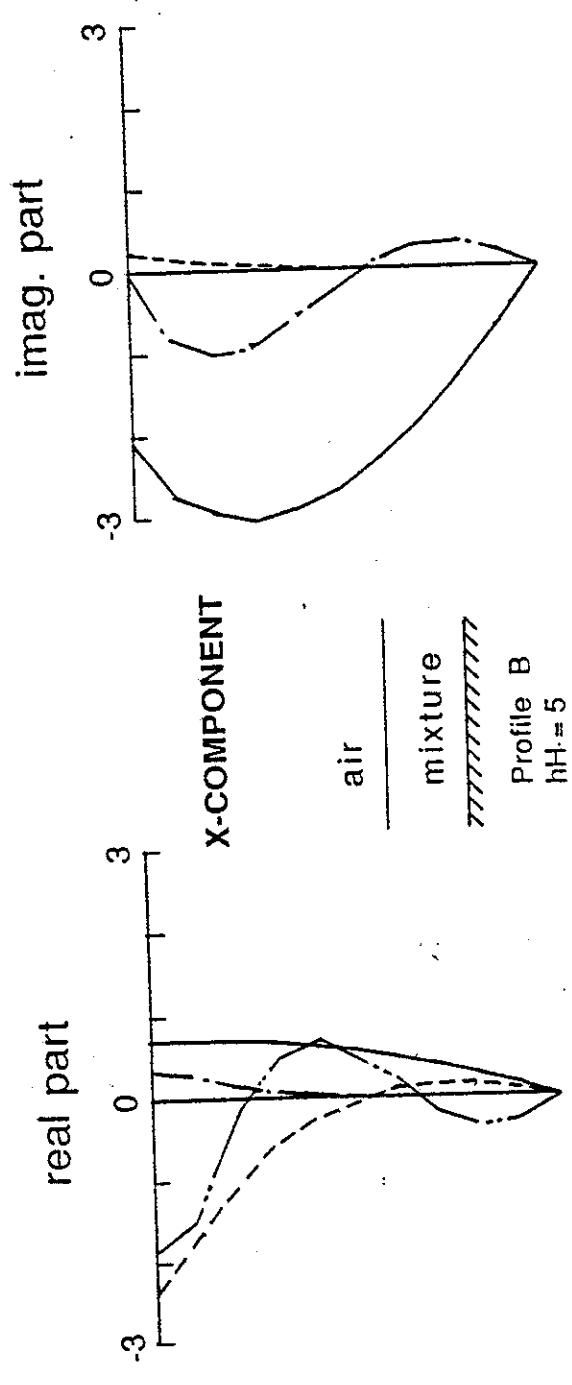


Fig. 6b Normalized displacement distributions for  $hH = 5$ , Profile B  
(a) air-solid

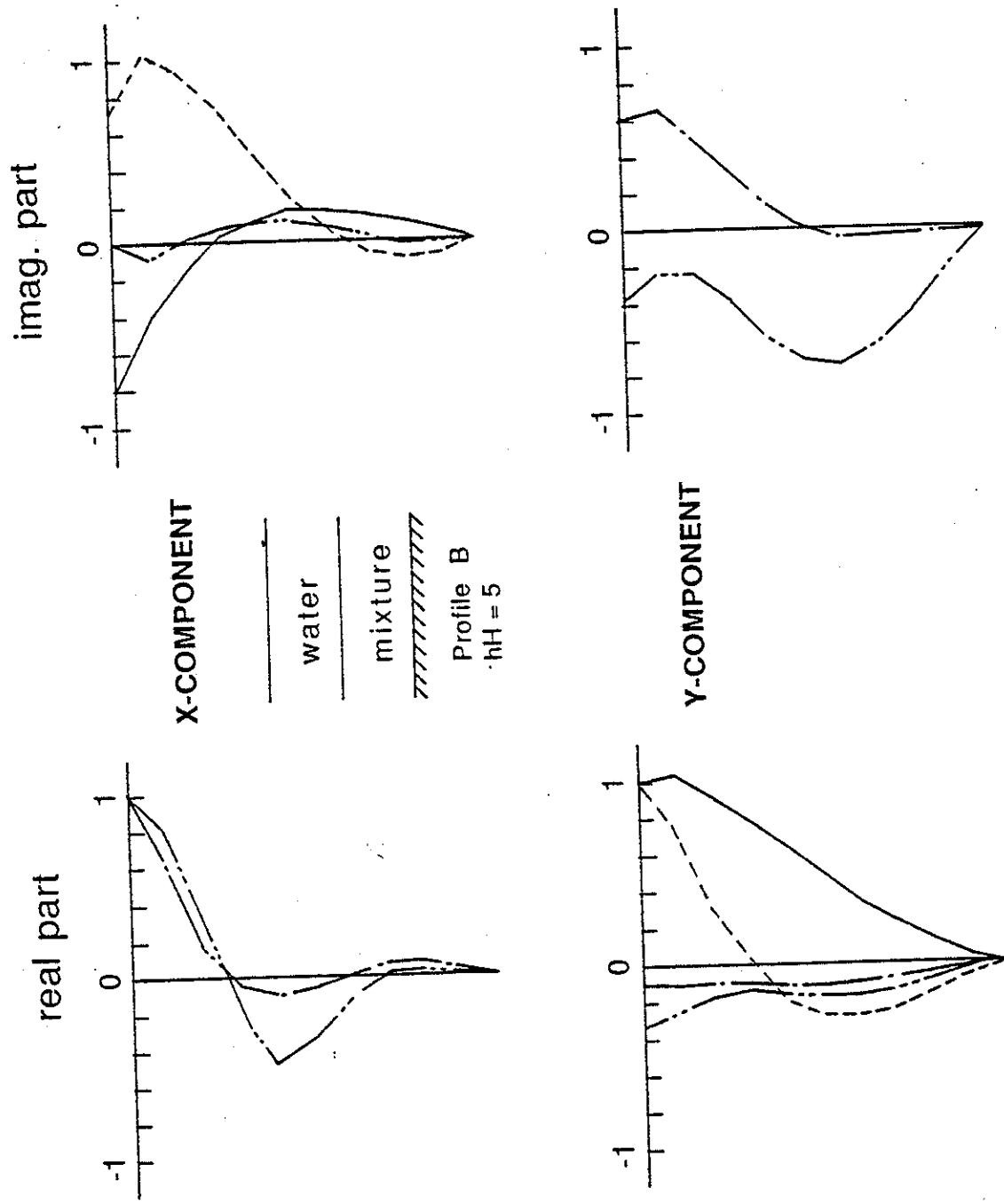


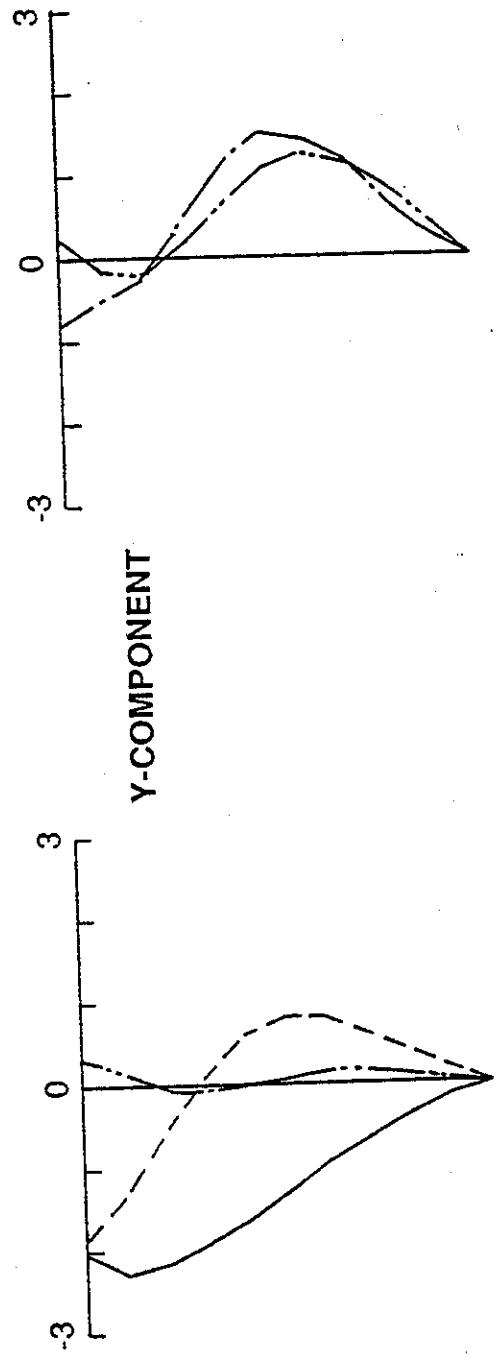
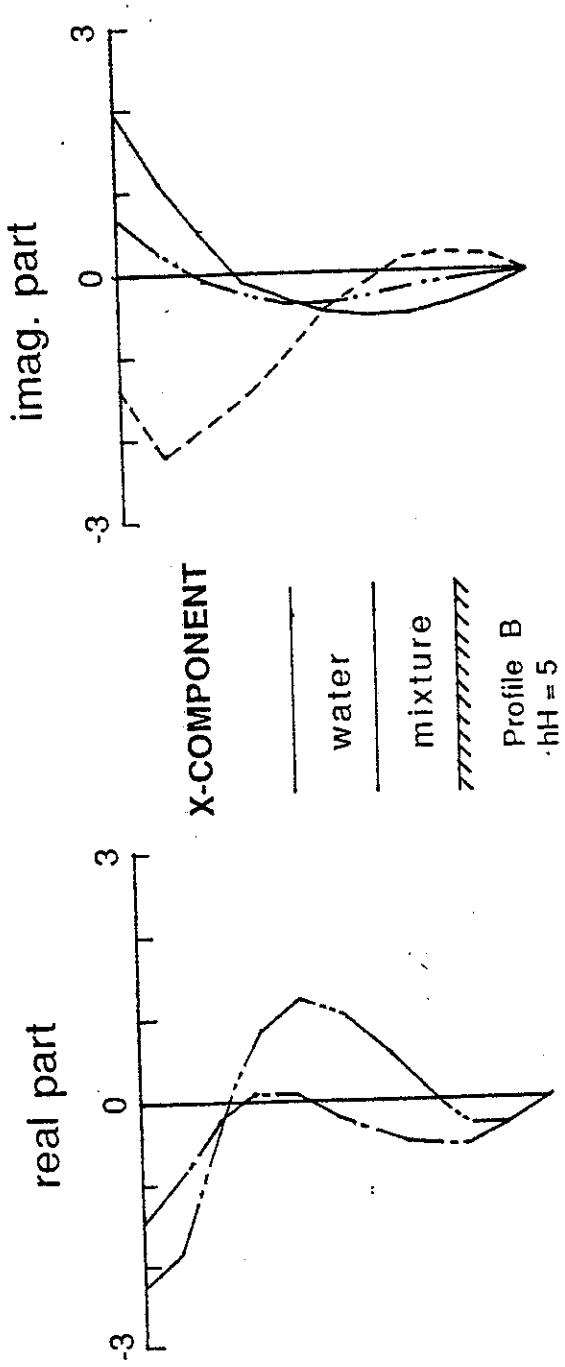
(b) air-mixture, solid frame



(c) air-mixture, fluid

(d) water-mixture, solid frame





(e) water-mixture, fluid

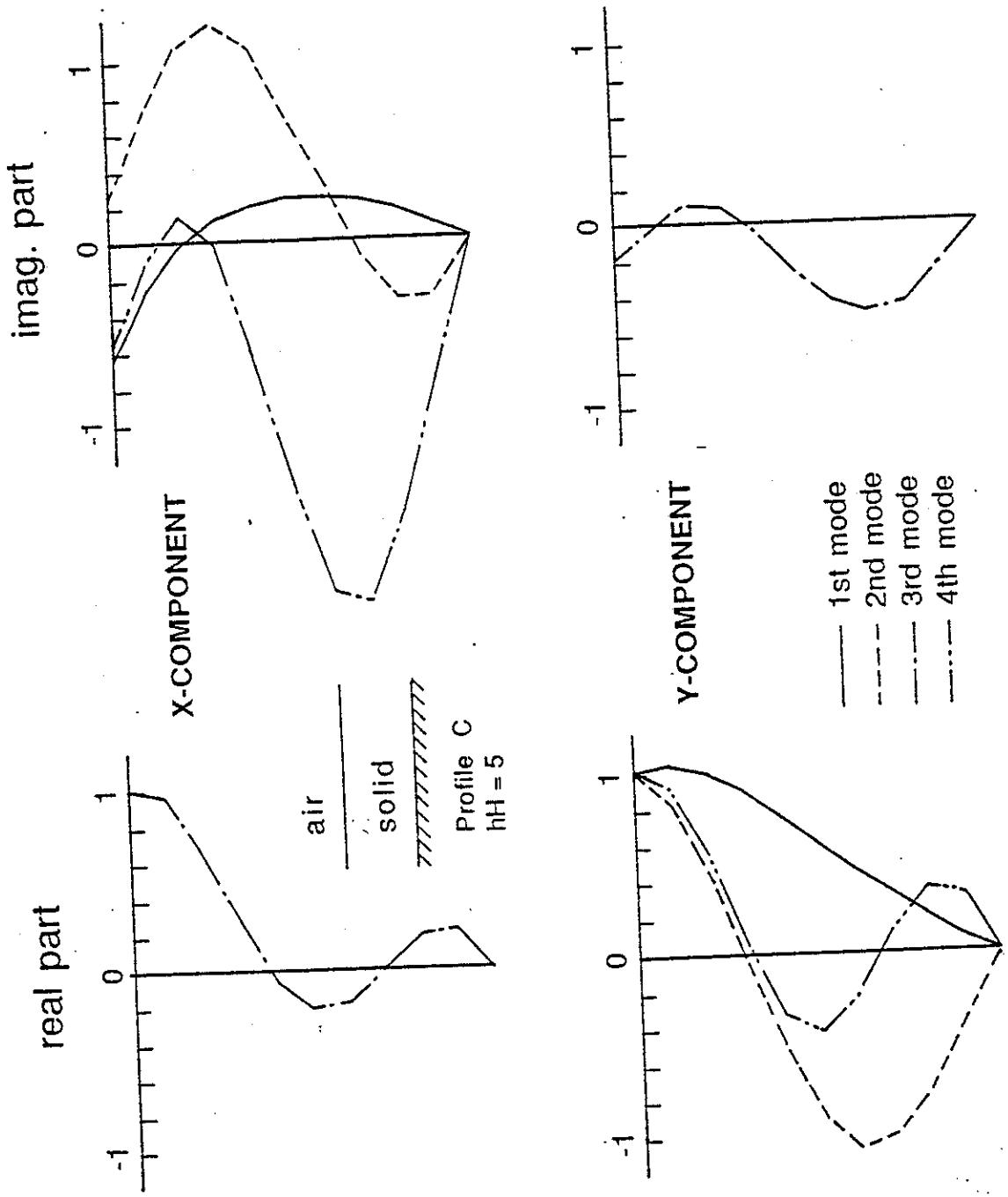
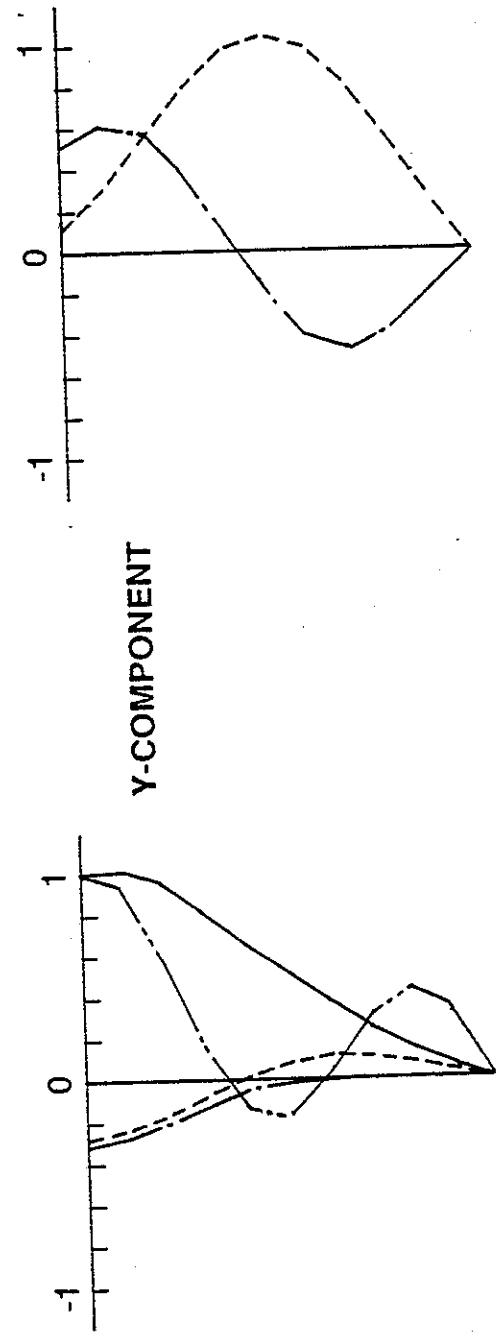
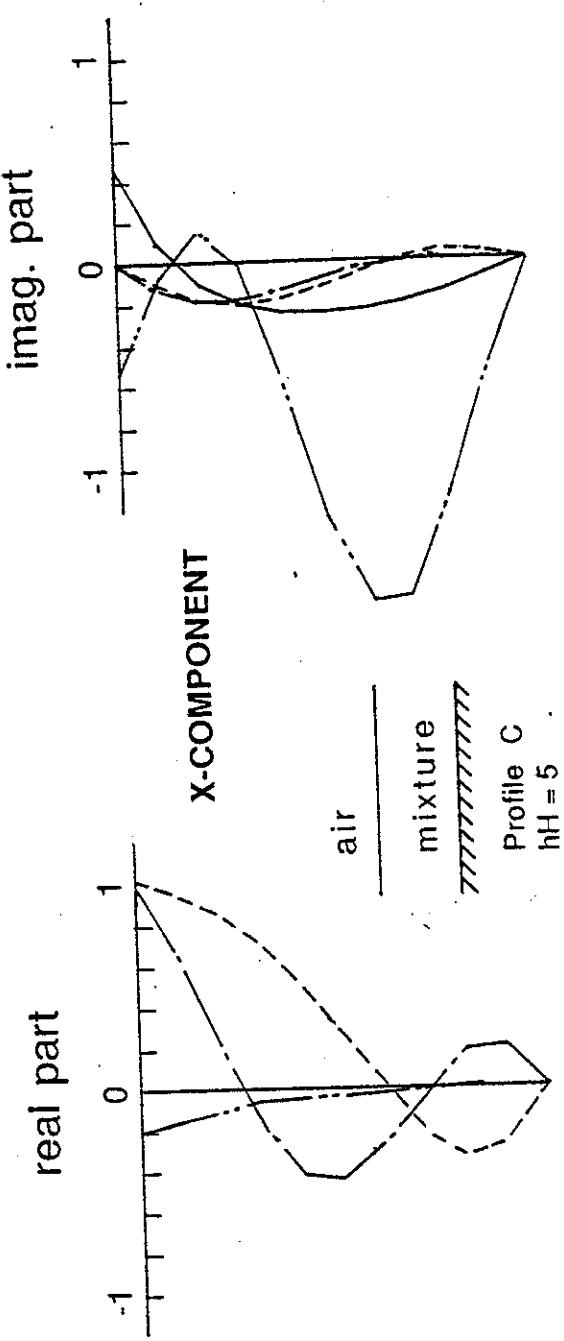
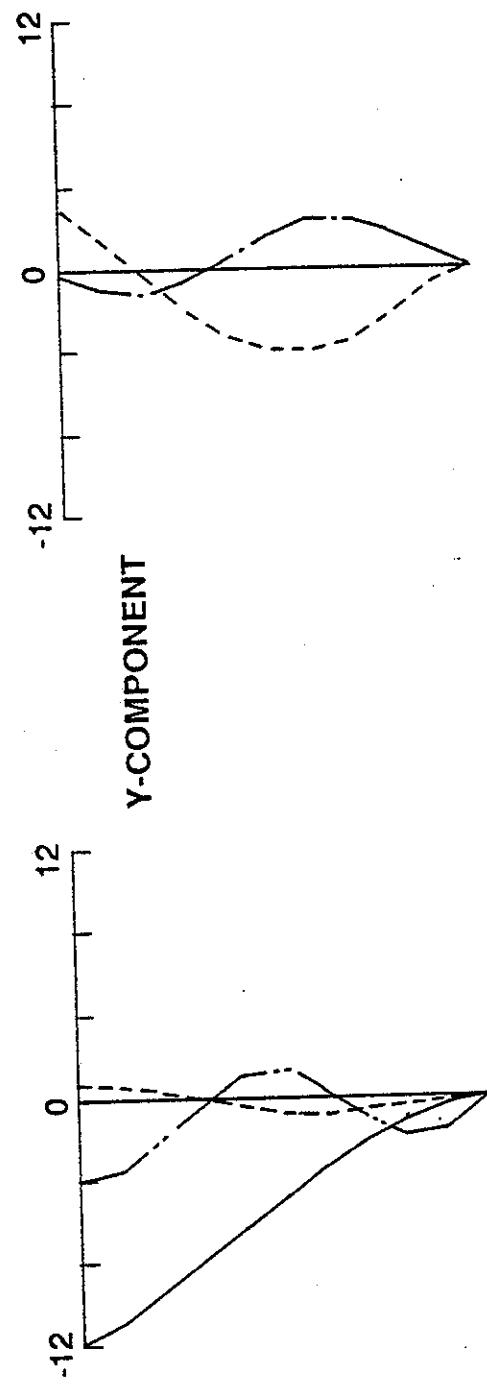
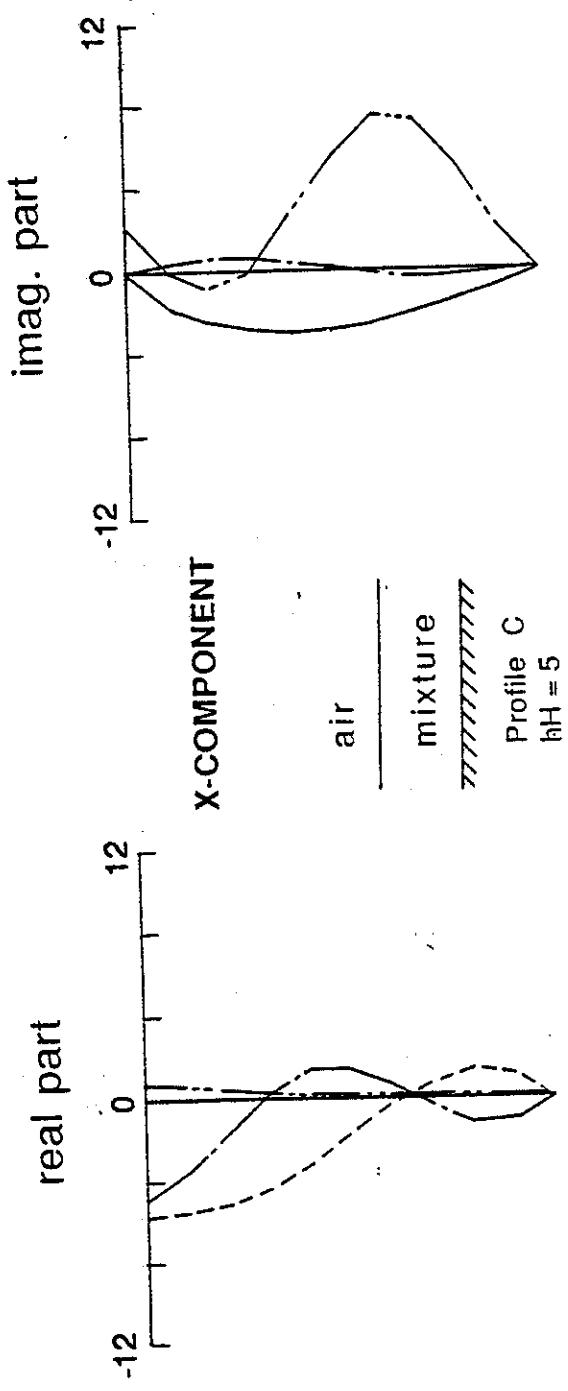


Fig. 6c Normalized displacement distributions for  $hH = 5$ , Profile C  
(a) air-solid



(b) air-mixture, solid frame



(c) air-mixture, fluid

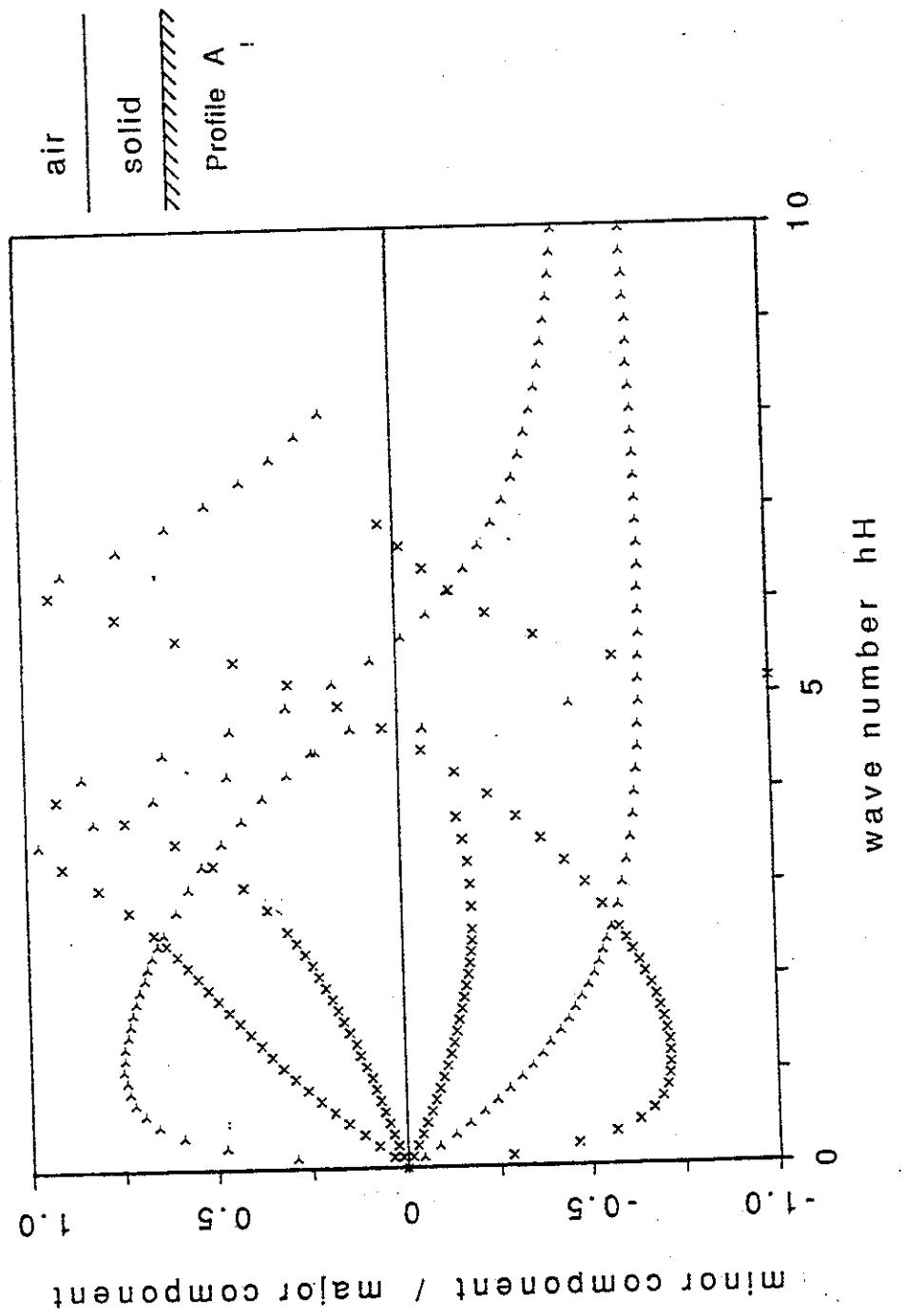
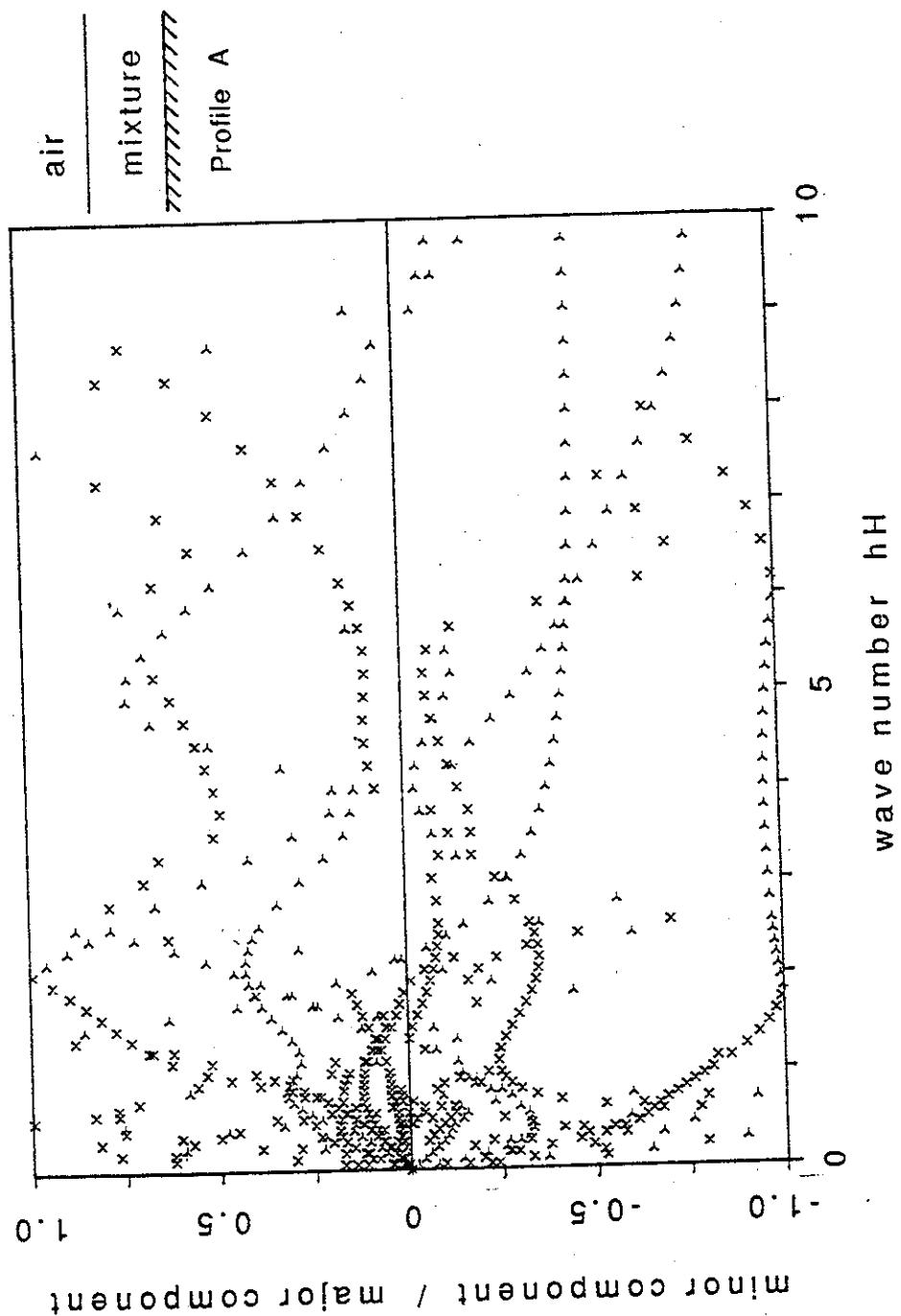
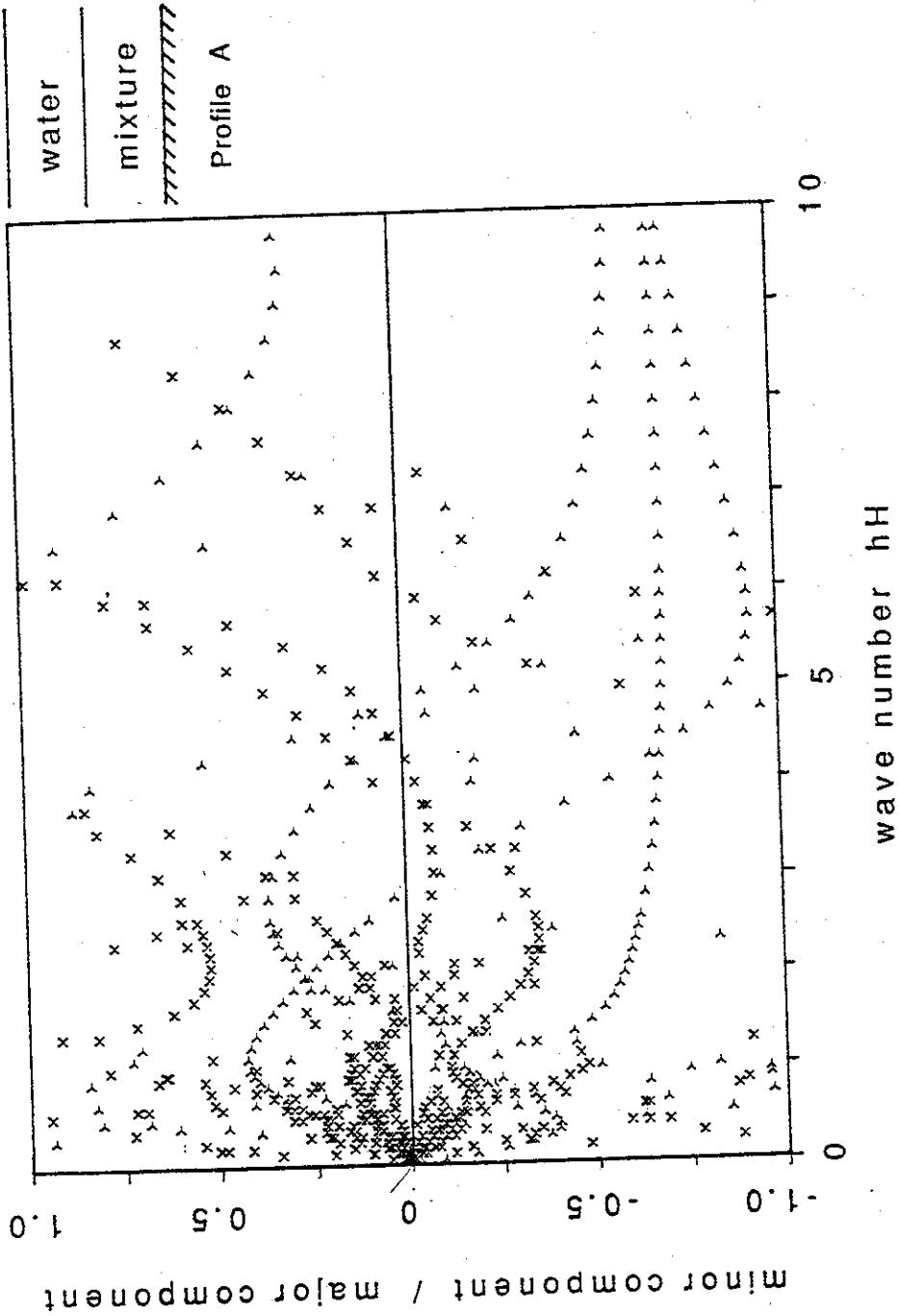


Fig. 7a Displacement amplitude ratios between two displacement components at the soil surface, Profile A  
 (a) air-solid



(b) air-mixture



(c) water-mixture

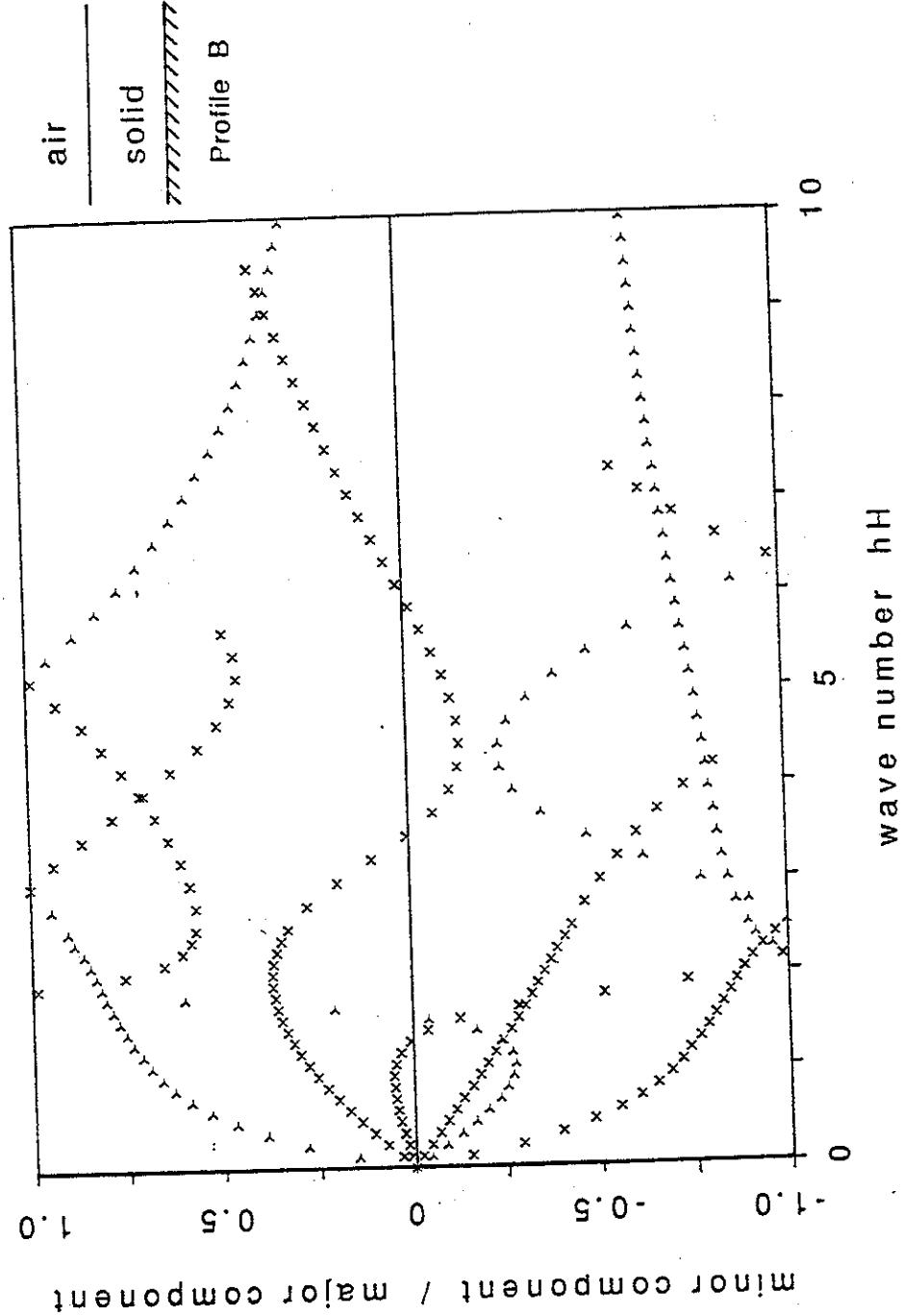
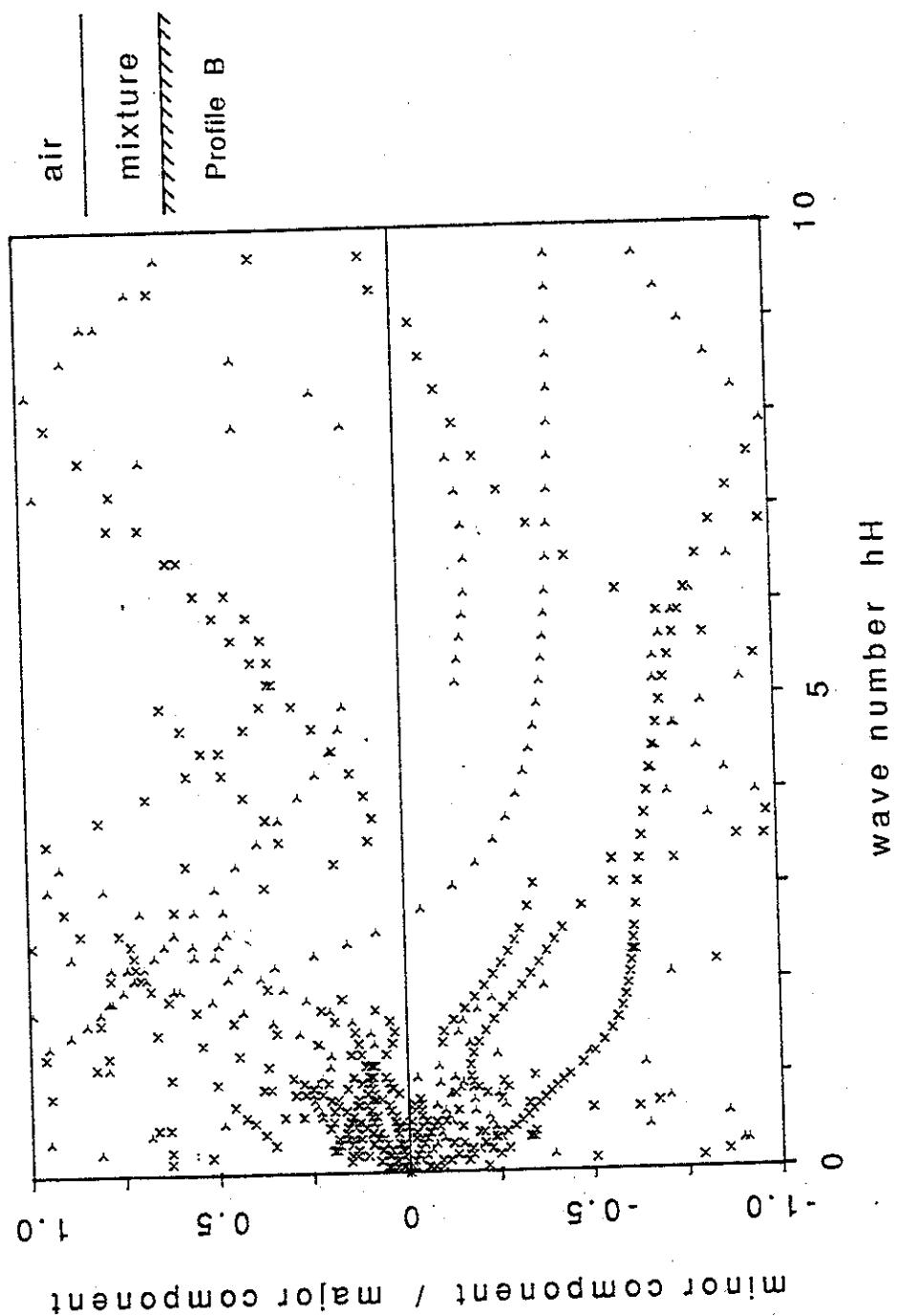
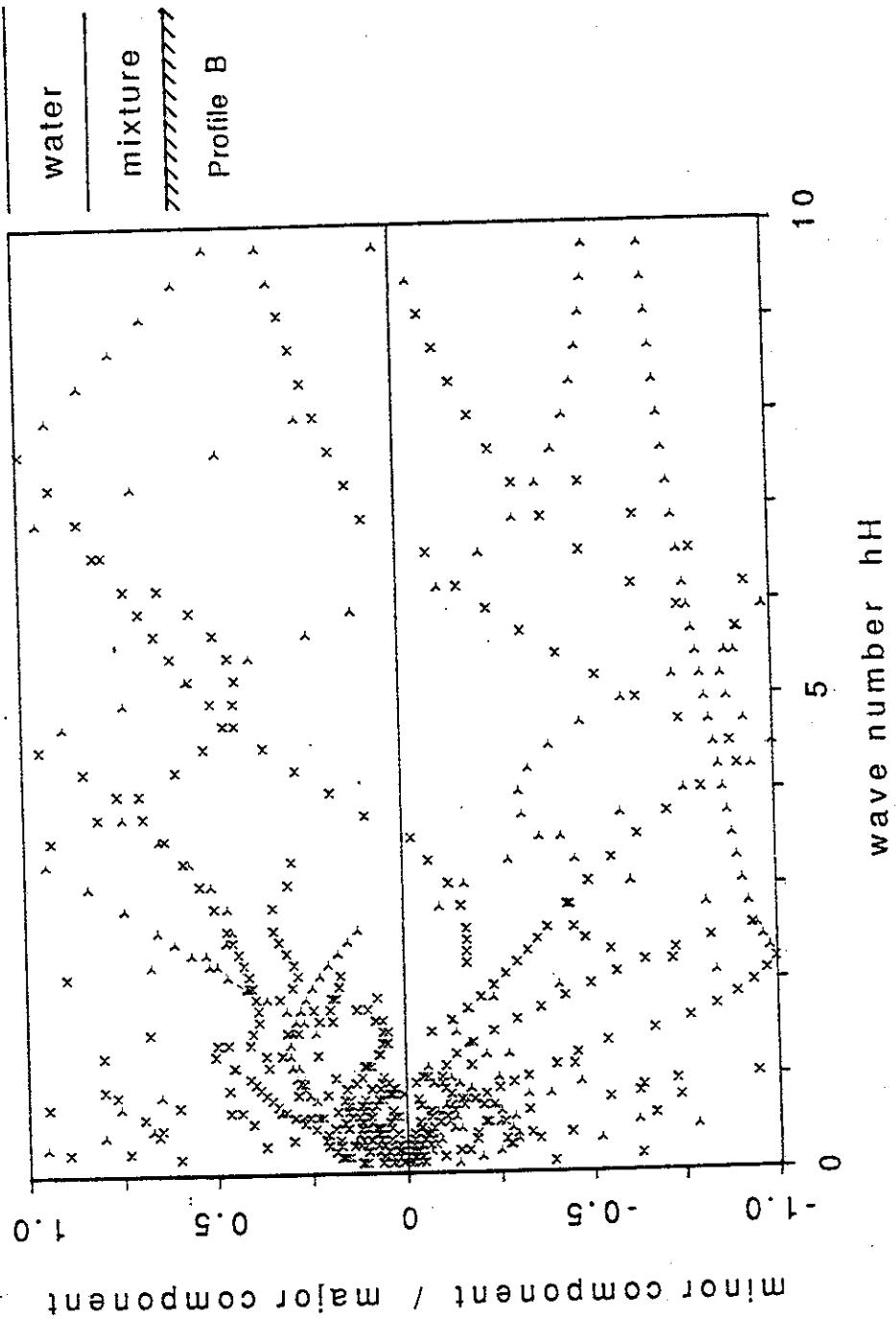


Fig. 7b Displacement amplitude ratios between two displacement components at the soil surface, Profile B  
 (a) air-solid



(b) air-mixture



(c) water-mixture

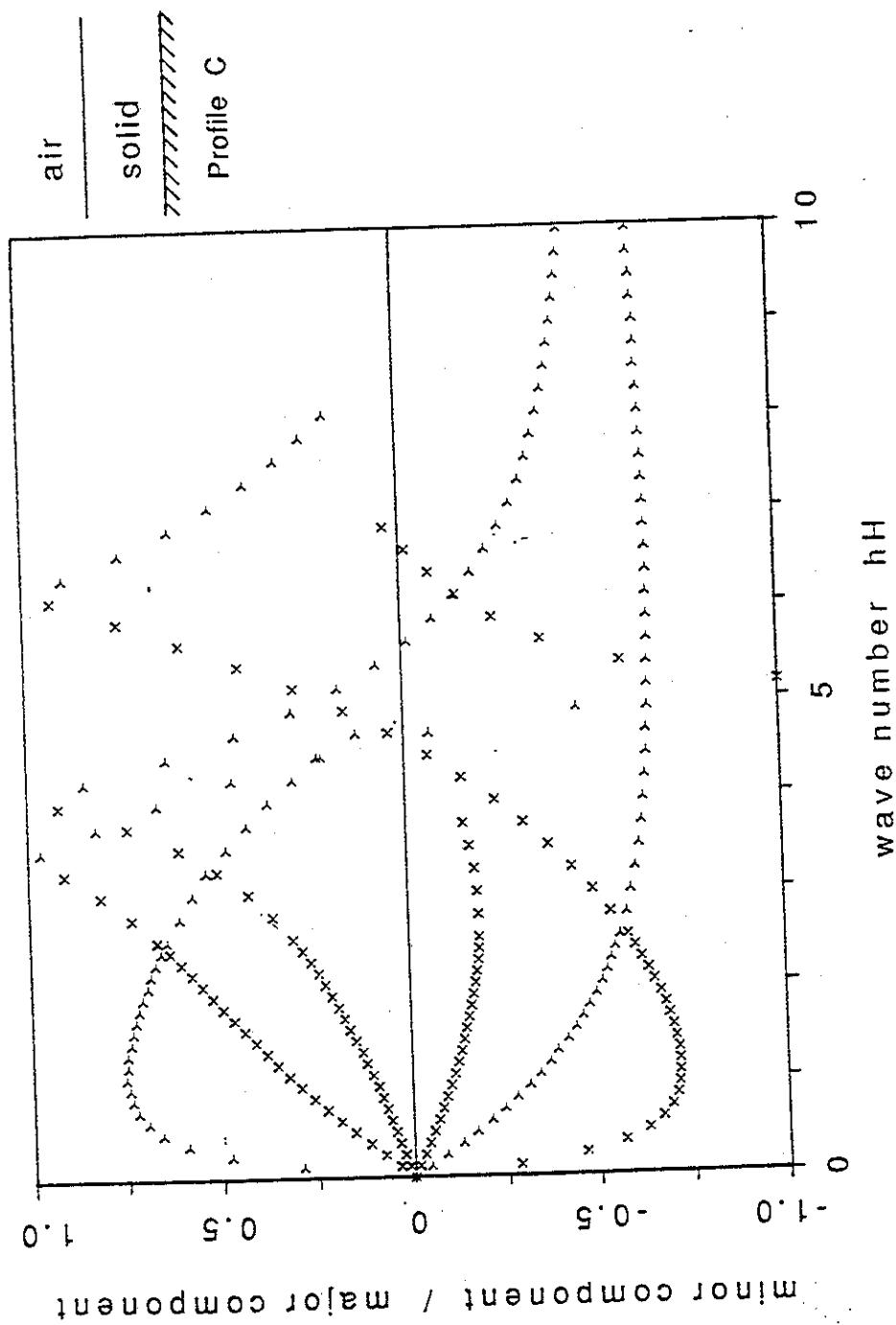
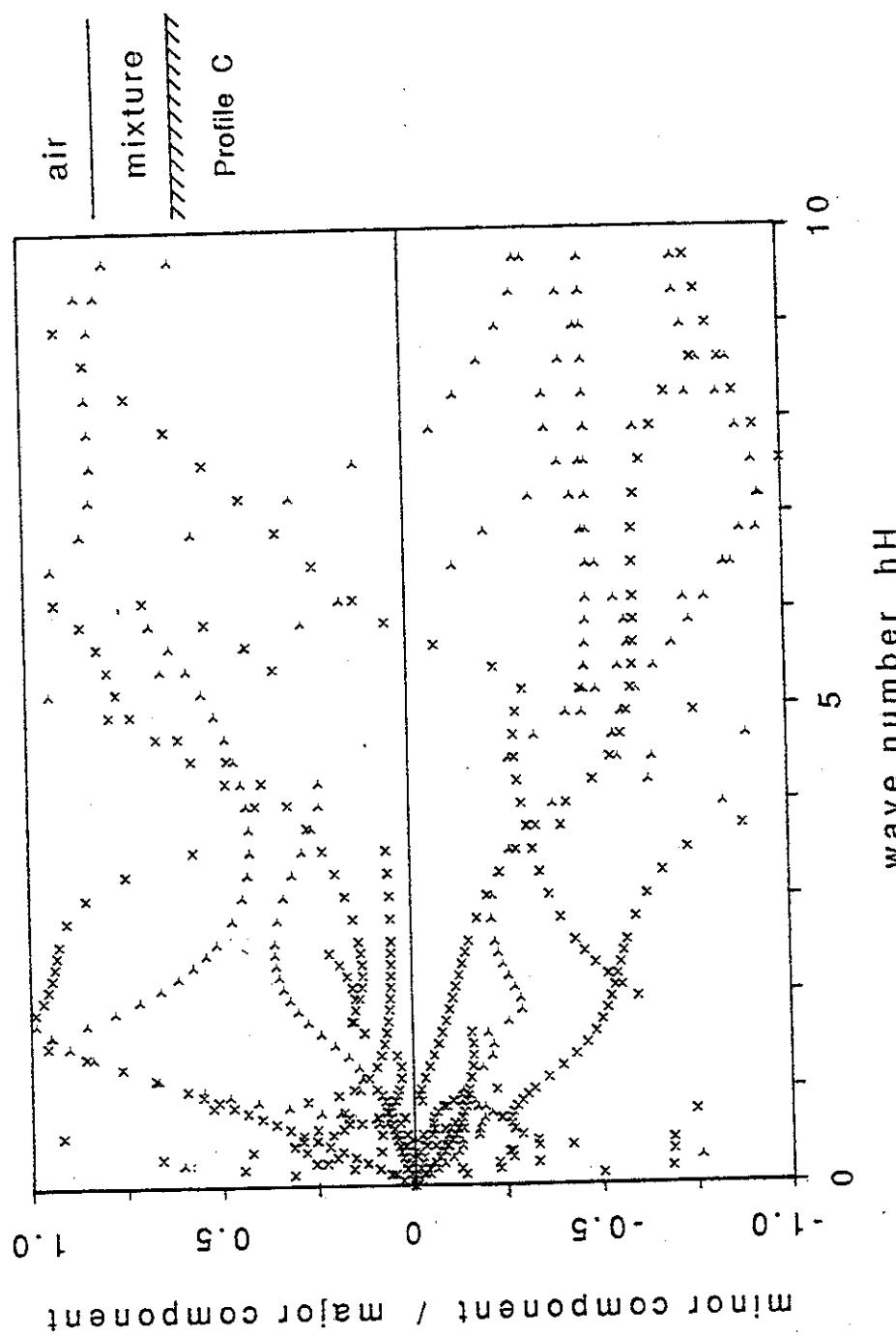


Fig. 7c Displacement amplitude ratios between two displacement components at the soil surface, Profile C  
 (a) air-solid

(b) air-mixture



## APPENDIX A

### DERIVATION OF GROUP VELOCITIES

Consider a solution,  $\omega$  and  $\mathbf{U}$ , to the eigenvalue problem in Eq. 19 and also slightly different number  $h+dh$ . The latter solution satisfies the equation:

$$[\mathbf{K} + \frac{\partial}{\partial h} \mathbf{K} dh - (\omega^2 + d\omega^2) \mathbf{M} + i(\omega + d\omega) \mathbf{C}] (\mathbf{u} + d\mathbf{u}) = 0 \quad (\text{A.1})$$

Using 19 and neglecting the small terms of the second order yields

$$[\frac{\partial}{\partial h} \mathbf{K} dh - d\omega^2 \mathbf{M} + i d\omega^2 \mathbf{M} + i d\omega \mathbf{C}] \mathbf{u} + [\mathbf{K} - \omega^2 \mathbf{M} + i \omega \mathbf{C}] d\mathbf{u} = 0 \quad (\text{A.2})$$

Premultiplying Eq. A.2 by  $\mathbf{U}^T$  leads to

$$\mathbf{u}^T \frac{\partial}{\partial h} \mathbf{K} \mathbf{u} dh + \mathbf{u}^T [\mathbf{K} - \omega^2 \mathbf{M} + i \omega \mathbf{C}] d\mathbf{u} = d\omega \mathbf{u}^T [2\omega \mathbf{M} - i \mathbf{C}] \mathbf{u} \quad (\text{A.3})$$

By taking into account the matrices  $\mathbf{K}$ ,  $\mathbf{M}$ , and  $\mathbf{C}$  are symmetric, the group velocity can be written as

$$V_g = \frac{d\omega}{dh} = \frac{\mathbf{U}^T \frac{\partial}{\partial h} \mathbf{K} \mathbf{U}}{\mathbf{U}^T [2\omega \mathbf{M} - i \mathbf{C}] \mathbf{U}} \quad (\text{A.4})$$

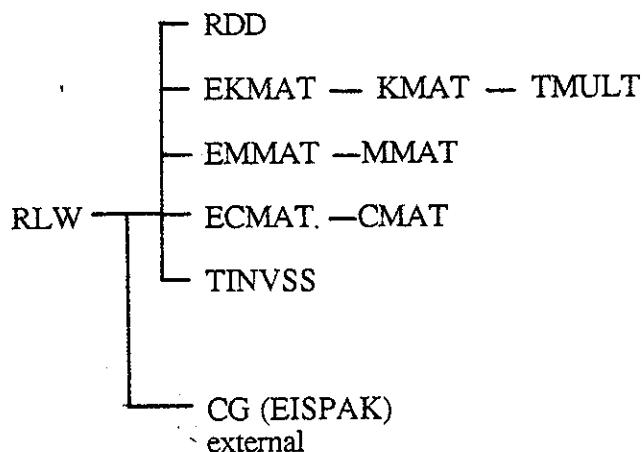
## APPENDIX B

### COMPUTER PROGRAM RLW

Finite element formulations developed in this study is coded in a computer program RLW. The listing of the program is attached at the end of this report.

#### 1. Program Structure

A flow chart of the program is shown in Fig. B.1. The program consists of several subroutines as follows:



where those subroutines are

- RDD: data input
- EKMAT: computation of global matrices A, B and G where  $K = h^2A + ihB + G$
- KMAT: computation of element matrices A, B and G for each element
- TMULT matrix multiplications ABC
- EMMAT: computation of global mass matrix M
- MMAT: computation of element mass matrix M for each element

ECMAT: computation of global damping matrix C  
CMAT: computation of element damping matrix C for each element  
TINVSS: matrix inversion  
CG: solution of eigenvalue problem

## 2. Input Information

---

TITLE  
(10A4)

---

TITLE = alphanumeric string

---

NMAT / NWATER /  
(2I5)

---

NMAT = number of sets of materials (< 5)

NWATER = number of elements for water

---

IMAT1 / NELMAT /  
(2I5)

---

---

THCK / GM / PO / ANR / AKS / PERM / ROES /  
(7F10.0)

---

Repeat those two cards as many as NMAT

NMAT1 = material set number

NELMAT = number of elements for material IMAT1

THCK = thickness of the layer (m)

GM = shear modulus (Kg/m<sup>2</sup>)

PO = Poisson's ratio

ANR = porosity

AKS = bulk modulus for solid grain, Ks (Kg/m<sup>2</sup>)

PERM = permeability (m/sec)

ROWS = mass density for solid phase (kg/m<sup>3</sup>)

---

AKF / ROW /  
(2F10.0)

---

AKF = bulk modulus of fluid, Kf (Kg/m<sup>2</sup>)

ROWF = mass density for fluid phase (Kg/m<sup>3</sup>)

---

NAK1 / IAT /  
(2I5)

---

NAK1 = number of wave numbers to be used in calculation

IAT = 0 (Wave numbers are chosen automatically)  
= 1 (Wave numbers are designated by the following cards)

---

AK1DAT(I) I=1, NAK1  
(7F10.0)

---

Those cards are not needed when IAT = 0

AK1DAT = wave number to be used

---

OUTFL /  
(A30)

---

OUTFL = file name to be used for output

### 3. Example

#### (a) Conditions considered

Example case considered is shown as follows.

$$v = 0.25 \quad \gamma_f = 1000 \text{ kg/m}^3 \quad K_s = \infty$$

$$\gamma_s = 2000 \text{ kg/m}^3 \quad K_f = 2208000000 \text{ kg/m}^2$$

depth	element	material		
0 (m)		G(kg/m <sup>2</sup> )	k(m/sec)	n
5	1			
10	2	4000000	0.004	0.45
15	3			
27	4			
40	5	5500000	0.001	0.33

(b) Inputs

The input listing for program RLW is

```
TEST 2 LAYERES, 5 ELEMENTS
 2   0
 1   3
 1   15.0 4000000.0    0.25    0.45    0.0    0.004    2000.0
 2   2
 2   25.0 5500000.0    0.25    0.33    0.0    0.001    2000.0
 2.208E9 1000.0
 2   1
 2   0.0    0.02
TEST_TLAY.DUT
```

(c) Outputs

The outputs by the computer program RLW are given in the following pages:

\*\*\*\*\*  
PROGRAM RLW  
Rayleigh wave in layered  
media of two-phase material  
\*\*\*\*\*

## title = TEST 2 LAYERS, 5 ELEMENTS

number of materials = 2

H= 40.000(m)

d= 0.000(m)

total number of elements = 5

average of Vs = 82.021(m/sec)

rho(fliuid)= 1000.000(kg/m\*\*3)

Kf= 2208000000.000(kg/m\*\*2)

material No.	$\rho$ (kg/m**2)	$\eta$	$\nu$	n	$K_E$ (kg/m**2)	$V_s$ (m/sec)	t (m)
1	4000000.0	0.250	0.450	inf.	0.004000	15.000	
2	5500000.0	0.250	0.330	inf.	0.001000	25.000	

material No.	$\lambda$ (kg/m**2)	lambda elements	number of elements	slab Q	$\rho$ (solid)	$\rho$ (solid)	$V_s$ (m/sec)
1	4000000.0	3	1.000	4906665984.0	2000.000	188.776	
2	5500000.0	2	1.000	6690909184.0	2000.000	200.559	

node	depth	element	material
1	0.000	-----	
2	5.000	-----	
3	10.000	-----	
4	15.000	-----	
5	27.500	-----	
6	40.000	-----	

h= 0.000

hH= 0.000

mode	$i*\Omega M_E(\text{Re})$	$i\Omega$	$i\Omega M_E(\text{Im})$	$i\Omega M_E(\text{Re})/V_s$	$-Re(i\Omega M_E)/i\Omega \omega_E$
1	-1.1027	0.0000	0.5378	1.0000	
2	-1.1025	0.0000	0.5377	1.0000	
3	-1.1075	0.0000	0.5401	1.0000	
4	-2.7706	0.0000	1.3512	1.0000	
5	-3.2673	0.0000	1.5934	1.0000	
6	-1.1636	-10.8170	5.3057	0.1070	
*	7	-1.1636	10.8170	5.3057	0.1070

8	-0.2709	4.0240	0.0328
9	-1.3407	-30.6865	14.9795
*	10	0.2709	8.2468
*	11	-1.3407	8.2468
12	-0.3260	30.6865	14.9795
13	-1.3261	-24.0352	11.7226
*	14	-0.3260	24.0352
*	15	-1.3261	55.6312
16	-0.3276	-43.3012	27.1380
17	-0.3276	43.3012	21.1178
18	-0.9249	-99.7480	27.1380
19	-0.9249	99.7480	21.1178
20	-0.2753	-74.9353	21.1178
21	-0.2753	74.9353	21.1178
22	-0.7218	-1.64.0244	79.9924
23	-0.7218	1.64.0244	79.9924
24	-0.2322	-120.6559	58.8417
25	-0.2322	120.6559	58.8417
26	-0.1230	-264.4764	128.9801
27	-0.1230	264.4764	128.9801
28	-0.1552	-777.9314	379.3824
29	-0.1552	777.9314	379.3824
30	-0.1459	-1393.2178	679.4459
31	-0.1459	1393.2178	679.4459
32	-0.1167	-2378.6860	1160.0400
33	-0.1167	2378.6860	1160.0400
34	-0.0893	-3783.4867	1845.1348
35	-0.0893	3783.4867	1845.1348
36	0.0000	0.0000	0.0000
37	0.0000	0.0000	0.0000
38	0.0000	0.0000	0.0000
39	0.0000	0.0000	0.0000
40	0.0000	0.0000	0.0000

MODE=	7	Ux(r)	Ux(th)	Uy(r)	Uy(th)	Wx(r)	Wx(th)	Wy(r)	Wy(th)	Wx(r)	Wx(th)	Wy(r)	Wy(th)
NODE													
J= 1	1.000	0.000	1.000	0.000	1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
J= 2	0.966	-0.230	0.978	-0.111	0.140	-166.357	0.978	-0.111	0.140	0.944	0.944	0.459	-0.111
J= 3	0.874	-1.090	0.913	-0.465	1.594	1.601	0.913	-0.466	1.601	0.902	0.902	0.807	-0.466
J= 4	0.731	-2.407	0.808	-1.140	1.767	15.395	0.808	-1.140	15.395	0.762	0.762	-0.400	-1.140
J= 5	0.388	-4.027	0.475	-2.261	1.828	21.987	0.475	-2.261	21.987	0.429	0.429	0.579	-2.262
ratio	0.0000	-17.1009	1.0000	0.0000	0.0000	165.1928	1.0018	179.9918	1.0018	10.7103	0.4514	0.0000	-172.6141

MODE=	14	Ux(r)	Ux(th)	Uy(r)	Uy(th)	Wx(r)	Wx(th)	Wy(r)	Wy(th)
J= 1	1.000	0.000	1.000	0.000	1.000	0.000	0.000	1.000	0.000
J= 2	0.810	-0.106	0.566	-44.620	0.817	-0.218	0.542	-67.109	
J= 3	0.312	-0.888	0.496	-79.509	0.296	-0.195	0.638	-106.466	
J= 4	0.304	-178.629	0.608	-52.358	0.241	-174.743	0.677	-69.543	
J= 5	0.842	-179.649	0.583	-74.005	0.610	-174.530	0.746	-85.210	
ratio	1.0000	0.0000	0.0000	-16.2638	0.4487	-177.3397	0.0000	165.7875	
h=	0.020								
hh=	0.800								
MODE	i*OMEGA (Re)	(Im)	!omega1!*H/Vs	-Re(i*omega1)/!omega1					
1	-1.1965	-7.0067	3.4665	0.1683					
2	-1.1692	-10.6189	5.2095	0.1094					
3	-0.8453	-13.8685	6.7760	0.0608					
4	-0.6779	-18.7047	9.1279	0.0362					
*	-1.1965	7.0067	3.4665	0.1683					
*	-1.1692	10.6189	5.2099	0.1094					
*	-0.8453	13.8685	6.7760	0.0608					
*	-0.6779	18.7047	9.1279	0.0362					
*	-1.2414	-27.7842	13.5633	0.0446					
10	-1.2708	-31.1074	15.1832	0.0408					
11	-1.2414	27.7842	13.5633	0.0446					
12	-1.2708	31.1074	15.1832	0.0408					
13	-0.1998	-38.0064	18.5353	0.0053					
14	-1.0348	-43.5038	21.2220	0.0238					
15	-0.1998	38.0064	18.5353	0.0053					
16	-1.0348	43.5038	21.2220	0.0238					
17	-1.3154	-55.8178	27.2288	0.0236					
18	-1.3154	55.8178	27.2288	0.0236					
19	-0.7414	-70.2833	34.2777	0.0105					
20	-0.7414	70.2833	34.2777	0.0105					
21	-0.1641	-77.7287	37.9069	0.0021					
22	-0.1641	77.7287	37.9069	0.0021					
23	-0.9217	-99.8314	48.6880	0.0092					
24	-0.9217	99.8314	48.6880	0.0092					
25	-0.0505	-110.8354	54.0523	0.0005					
26	-0.0505	110.8354	54.0523	0.0005					
31	-0.1350	-290.0145	141.4346	0.0005					
32	-0.1350	290.0145	141.4346	0.0005					
33	-0.1547	-784.5720	382.6209	0.0002					
34	-0.1547	784.5720	382.6209	0.0002					
35	-0.1459	-1396.8239	681.2046	0.0001					
36	-0.1459	1396.8239	681.2046	0.0001					
37	-0.1167	-2380.2740	1160.8145	0.0000					
38	-0.1167	2380.2740	1160.8145	0.0000					
39	-0.0893	-3784.3664	1845.5649	0.0000					

MODE= 40		-0. 0893		3784. 3684		1845. 5649		0. 0000	
NODE		5		Ux(r)		Ux(th)		Uy(r)	
J= 1	1. 000	0. 000	1. 000	0. 000	0. 000	0. 000	0. 000	0. 000	0. 000
J= 2	0. 986	-0. 396	0. 722	-3. 465	0. 967	-0. 429	0. 717	-2. 226	0. 000
J= 3	0. 923	-1. 410	0. 445	-10. 539	0. 855	-1. 681	0. 459	-6. 031	0. 000
J= 4	0. 817	-3. 218	0. 210	-29. 278	0. 673	-4. 299	0. 259	-11. 897	-5. 409
J= 5	0. 478	-6. 257	0. 201	-154. 820	0. 394	-7. 171	0. 044	-134. 458	-28. 747
ratio	1. 0000	0. 0000	0. 1230	-99. 2146	1. 4999	-179. 1541	0. 2915	85. 6076	1. 301
NODE		6		Ux(r)		Ux(th)		Uy(r)	
J= 1	1. 000	0. 000	1. 000	0. 000	0. 000	1. 000	0. 000	0. 000	0. 000
J= 2	0. 750	-0. 865	0. 992	-0. 165	0. 705	-2. 834	0. 988	-0. 386	0. 000
J= 3	0. 451	-2. 466	0. 935	-0. 577	0. 326	-10. 222	0. 928	-0. 958	-16. 582
J= 4	0. 154	-7. 320	0. 831	-1. 301	0. 065	-126. 870	0. 821	-1. 754	-26. 828
J= 5	0. 237	-178. 767	0. 484	-2. 475	0. 416	-171. 850	0. 471	-2. 988	-18. 546
ratio	0. 4757	-93. 0719	1. 0000	0. 0000	0. 4470	80. 9722	1. 0505	-179. 4379	1. 577
NODE		7		Ux(r)		Ux(th)		Uy(r)	
J= 1	1. 000	0. 000	1. 000	0. 000	1. 000	0. 000	0. 000	0. 000	0. 000
J= 2	1. 175	-11. 335	0. 899	1. 318	0. 799	-1. 315	0. 615	-5. 409	0. 000
J= 3	1. 718	-30. 205	0. 770	2. 469	0. 270	-10. 797	0. 325	-16. 582	-18. 546
J= 4	2. 423	-42. 073	0. 659	12. 329	0. 415	-168. 864	0. 213	-26. 828	-32. 550
J= 5	2. 334	-50. 890	0. 614	52. 336	0. 513	-178. 171	0. 188	-33. 302	-33. 302
ratio	1. 0000	0. 0000	0. 3191	-49. 0326	4. 1230	-56. 9267	1. 1227	-149. 2829	0. 0569
NODE		8		Ux(r)		Ux(th)		Uy(r)	
J= 1	1. 000	0. 000	1. 000	0. 000	1. 000	0. 000	0. 000	0. 000	0. 000
J= 2	0. 927	-1. 016	0. 878	0. 058	0. 840	-2. 887	1. 172	-18. 541	0. 000
J= 3	0. 672	-4. 925	0. 740	-0. 094	0. 339	-19. 000	1. 301	-28. 747	1. 411
J= 4	0. 318	-117. 707	0. 618	-0. 168	0. 305	-149. 227	1. 411	-32. 550	1. 577
J= 5	0. 212	-159. 472	0. 387	0. 642	1. 080	172. 947	1. 577	-33. 302	1. 577
ratio	1. 0000	0. 0000	0. 3206	-87. 4384	0. 4443	-163. 4493	0. 0569	127. 3362	0. 0569

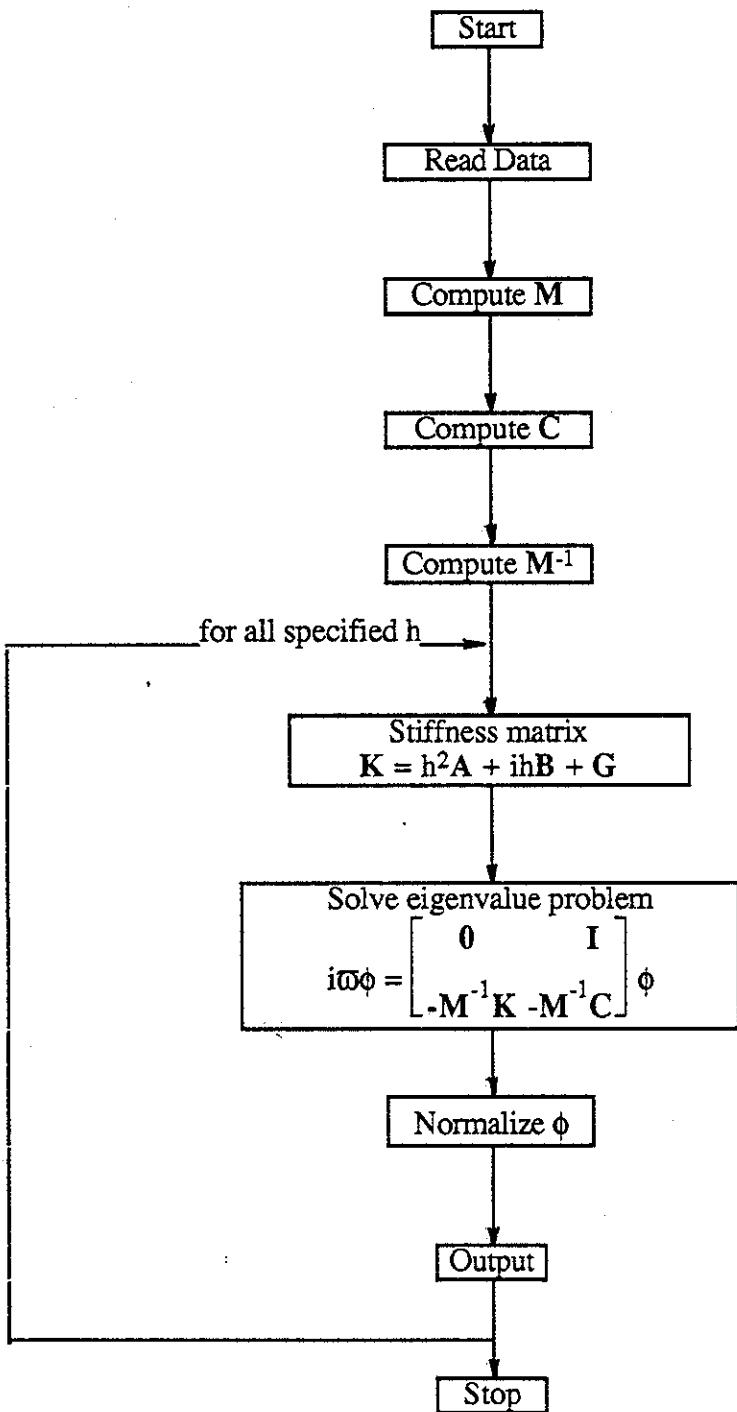


Fig. B.1 Flow chart of computer program RLW

**APPENDIX C**  
**LISTING OF COMPUTER PROGRAM RLW**

Salford University FTN77 Ver. 230dS LISTING INTS NOMAP NOCHECK NOBIC LOGS DYNM OFFSET NDANSI NODBUG NOPAGE1THROW NOFRN

COMPILER OPTIONS: CENGR>AKIRA>SEC>RLW. FTN77  
 FPN NOLUNFREC NO\_OPTIMIZE NO\_IMPURE

```

0001 C***** PROGRAM RLW
0002 C-----
0003 C ++++++
0004 C + Rayleigh Wave in Solids Consisting of Two Phases
0005 C + based on program codes, RAYL15. F77 & RAYLIB. F77
0006 C + (MAY 1988)
0007 C ++++++
0008 C ++++++
0009 PARAMETER (NEL=17, MPR=5, MKG=700, MSZ=4*MEL)
0010 COMMON /ACE/ACE(B,B)
0011 COMMON /ABQ/AMAE(B,B), AMB(MSZ,MSZ), AMG(MSZ,MSZ)
0012 COMMON /ABCE/AMAE(B,B), AMBE(B,B), AMGE(B,B), AMME(B,B)
0013 COMMON /AC/AC(MSZ,MSZ)
0014 COMMON /GM/GM(MPR), PERM(MPR), P0(MPR), ALFA(MPR), ANR(MPR), RM(MPR)
0015 R, Q(MPR), AKF, AKS(MPR), NMAT, LMAT(MEL), NMAT
0016 COMMON /ROW/ROWS(MPR), ROWF, ROW(MPR)
0017 COMMON /HT/AK1,B(MEL), H,NELMAT(MPR), THCK(MPR)
0018 COMMON /NODE/NNODE, NODE2, NODE4
0019 COMMON /SLR/SL(2*MSZ,2*MSZ)
0020 COMMON /AAM/AAR(2*MSZ,2*MSZ), AAI(2*MSZ,2*MSZ)
0021 COMMON /CGM/WR(2*MSZ), WJ(2*MSZ), FW1(2*MSZ), FW2(2*MSZ)
0022 COMMON /ZVS/ZR(2*MSZ,2*MSZ), ZV1(2*MSZ,2*MSZ)
0023 COMMON /LPO/ILPQUT, ILPT
0024 COMMON /AMK/AMK(MSZ,MSZ)
0025 COMMON /AMM/AMM(MSZ,MSZ)
0026 COMMON /PLT/WVN0(MKG), ANDR(MKG), ANDI(MKG), WVN0G2(MKG), UDWN(MKG)
0027 DIMENSION 1W(2*MSZ), AMT(MSZ,MSZ), IPASS(2*MSZ), THSL(4), IC(MKG)
0028 DIMENSION V5(MPR), AKIDAT(100)
0029 COMPLEX SL
0030 COMPLEX SLS1, SLS2, SLS3, SLS4
0031 REAL*B, AAR, AAI, WR, WI, ZR, ZI, FW1, FW2, FW3
0032 CHARACTER 1PASS*1, INFL*30, OUTFL*30, FL11*30, TITLE(10)*4
0033 DATA QCNST/2, B/, FBAL/1, 0 /, P1/3, 14159265359/, ANDMX/12, 0/
0034 ILPQUT=0
0035
0036 C * INPUT INFORMATION
0037 C * INPUT INFORMATION
0038 C----- 40
0039 C----- :
0040 C----- TITLE
0041 C----- (10A4) One Card
0042 C----- C TITLE = An alphanumeric string
0043 C----- C
0044 C----- 5 10
0045 C----- NMAT ! NWATER !
0046 C----- C
0047 C----- C
0048 C----- C (215) One Card
0049 C----- C NMAT = number of the sets of the materials (i.e. different layers)
0050 C----- C NWATER = number of elements which are used as "water elements"
0051 C----- C
0052 C----- C

```

```

0053      5      10
C 1MAT1 ! NELMAT !
C 10      20      30      40      50      60      70
C THICK  ! GM   ! PG   ! ANR  ! AKS  ! PERM  ! ROWS !
C
C (215)          As many as required
C (7F10.0)
0062  C 1MAT1 = material set number
0063  C NELMAT = number of elements which is used to the material 1MAT1
0064  C THICK = thickness of the layer (m)
0065  C GM = shear modulus (kg/m**2)
0066  C PO = Poisson ratio, nyu
0067  C ANR = porosity, n
0068  C AKS = bulk modulus for solid grain, Ks (kg/m**2)
0069  C * If AKS .LE. 0.0 then Ks=infinity
0070  C PERM = permeability, k (m/sec)
0071  C * If PERM .LE. 0.0 then k=infinity (no damping)
0072  C ROWS = mass density for solid phase (kg/m**3)
0073
0074
0075      10      20
C
C AKF  ! ROWF  !
C
C (2F10.0)          One Card
0076  C AKF = bulk modulus for fluid, Kf (kg/m**2)
0077  C ROWF = mass density for fluid phase (kg/m**3)
0078
0079
0080
0081
0082
0083
0084
0085  C NAK1 : IAT !
0086
0087  C (215)          One Card
0088  C NAK1 = number of wave numbers to be used in calculation
0089  C IAT = 0 : wave numbers are chosen automatically
0090  C = 1 : wave numbers are designated by the following cards
0091
0092      10      20      30      60      70
0093  C AKIDAT(1) : AKIDAT(2) : AKIDAT(3) : ..... : AKIDAT(7) :
0094
0095  C AKIDAT(B) : AKIDAT(9) : ..... : AKIDAT(NAK1) :
0096
0097  C (7F10.0)          As many as required if IAT=1 ; not needed if IAT=0
0098  C AKIDAT = value of wave number used in the calculation (rad/m)
0099
0100
0101
0102
0103
0104
0105  C OUTFL  !          One Card
0106  C OUTFL = file name to be used for output list
0107
0108  WRITE(*,*), INPUT FILE NAME = ?
0109

```

AT 3  
AT 15

```

0110 READ(*, '(A30)') INFL          AT 35
0111 OPEN(2, FILE=INFL)           AT 6B
0112 CALL RDD(TITLE, NELM, ROWS, NAKF, NAK1, IAT, AK1DAT, OUTFL, FL11)
0113 CLOSE(2, STATUS='KEEP')
0114 WRITE(*, *) 'TITLE = ', TITLE
0115 C----- ROWF=ROWF/GCNST
0116 DO 42 IMAT=1, NMAT           AT 154
0117 ROWS(IMAT)=ROWS(IMAT)/GCNST
0118 RM(IMAT)=2.0*GM(IMAT)*PO(IMAT)/(1.0-2.0*PO(IMAT))
0119 01
0120 01
0121 01 IF (AKS(IMAT) .LE. 0.0) THEN << Ks is assumed to be infinite>>
0122 01
0123 02 ALFA(IMAT)=1.
0124 02 QIMAT=AKF/ANR(IMAT)
0125 02 ELSE
0126 02   ALFA(IMAT)=1.-AKD/AKS(IMAT)
0127 02   QIMAT=1./ ( ANR(IMAT)/AKF+(ALFA(IMAT)-ANR(IMAT))/AKS(IMAT) )
0128 02   IF (ALFA(IMAT) .LT. ANR(IMAT)) WRITE(*, *) 'WARNING life < n,
0129 02 ENDIF
0130 01 ROW(IMAT)=ANR(IMAT)*ROWF+(1.0-ANR(IMAT))*ROWS(IMAT)
0131 01 VS(IMAT)=SORT( GM(IMAT)/(1.0-ANR(IMAT))*ROWS(IMAT) )
0132 01 42 CONTINUE
0133 H=0.0
0134 DO 43 IELM=NWATER+1, NELM
0135 01 43 H=H+B(IELM)
0136 DO 44 IELM=1, NWATER
0137 d=d+B(IELM)
0138 01 44 d=d+B(IELM)
0139 VSA=0.0
0140 DO 45 IELM=NWATER+1, NELM
0141 01 A5 VSA=VSA+B(ILMAT(IELM))/VS(ILMAT(IELM))
0142 VSA=H/VSA
0143 C----- OPEN(10, FILE=OUTFL)
0144 OPEN(6, FILETYPE='TTY')
0145 NNODE=NNLM+1
0146 NNODE2=2*NNODE
0147 NOD4=A*NNODE
0148 NOD4A=A*(NNODE-1)
0149 C----- WRITE(10, '(//X, '*****')')
0150
0151 WRITE(10, '(//X, '*****')') PROGRAM RLW
0152 WRITE(10, '(//X, '*****')') 'Rayleigh wave in layered'
0153 WRITE(10, '(//X, '*****')') 'media of two-phase material'
0154 WRITE(10, '(//X, '*****')') 'title = ''1044'') (TITLE(1),I=1,10)
0155 WRITE(10, '(//X, '*****')') 'number of materials = ',13,') NMAT
0156 WRITE(10, '(//X, '*****')') 'H'
0157 WRITE(10, '(//X, '*****')') 'd'
0158 WRITE(10, '(//X, '*****')') 'total number of elements = ',13,') NELM
0159 WRITE(10, '(//X, '*****')') 'm/sec)', ')) VS
0160 WRITE(10, '(//X, '*****')') 'Average of VS = ',F10.3,'(kg/m**2)) ROWF*GCNST
0161 WRITE(10, '(//X, '*****')') 'rho-fluid)= ',F10.3,'(kg/m**2)) AKF
0162 IELFR=1
0163 WRITE(10, '(//X, '*****')') 'Kf= ',F15.3,'(kg/m**2))')
0164 DO 1120 IMAT=1, NMAT
0165 1120 CONTINUE
0166 01

```

```

0167      WRITE(10, '(/3X, ''material'',          ngy        ,''      ,'
0168      &           '       ,      Ks        ,''      ,''      t      ,'
0169      &           '       , No.      ,      k        ,''      ,''      ,'
0170      &           '       , (kg/m**2)  ,''      ,''      ,'
0171      &           '       , (kg/m**2)  ,''      ,''      ,'
0172      &           '       , (kg/m**2)  ,''      ,''      ,'
0173      DO 1130 IMAT=1,NMAT
0174      IF(AKS(IMAT).GT. 0.0) THEN
0175      WRITE(10, '(2X,15,F13.1,F10.3,F14.1,F10.6,F10.3)')
0176      & IMAT,GM(IMAT),PO(IMAT),ANR(IMAT),AKS(IMAT),PERM(IMAT),THCK(IMAT)
0177      ELSE
0178      WRITE(10, '(2X,15,F13.1,F10.3,F10.3,9X,A4,F10.6,F10.3)')
0179      & IMAT,GM(IMAT),PO(IMAT),ANR(IMAT),inf.,PERM(IMAT),THCK(IMAT)
0180      ENDIF
0181      01 1130 CONTINUE
0182      WRITE(10, '(/3X, ''material'',          lambda     ,''      ,'
0183      &           '       ,      Q        ,''      ,''      alfa     ,'
0184      &           '       , No.      ,      rho(solid),''      ,'
0185      &           '       , (kg/m**2)  ,''      ,''      ,'
0186      &           '       , (kg/m**2)  ,''      ,''      ,'
0187      &           '       , (kg/m**2)  ,''      ,''      ,'
0188      DD 1140 IMAT=1,NMAT
0189      01 1140 WRITE(10, '(2X,15,F13.1,F10.3,F14.1,F10.3,F10.3)')
0190      & IMAT,GM(IMAT),NELMAT(IMAT),ALFA(IMAT),G(IMAT),
0191      & ROWS(IMAT)*GCNET,VS(IMAT),
0192      C-----)
0193      WRITE(10, '(/)')
0194      WRITE(10, '( 3X, ''node'', 5X, ''depth'', 5X, ''element'',
0195      &           '       ,      5X, ''material'',/)')
0196      DEPTH=-d
0197      ILB=0
0198      IEB=1
0199      DD 1150 INODE=1,NNODE
0200      IF(INODE.GT. 1) DEPTH=DEPTH+B(INODE-1)
0201      01  IF(INODE.EQ. IEB) THEN
0202      02  IF(INODE.EQ. NWATER+1) OR. INODE.EQ. 1) THEN
0203      03  WRITE(10, '( 3X,13,F10.3,B(1H-),1H+,1B(1H-))') INODE,DEPTH
0204      03  ELSE
0205      03  WRITE(10, '( 3X,13,F10.3,B(1H-),1H+, B(1H-))') INODE,DEPTH
0206      03  ENDIF
0207      02  ILB=ILB+1
0208      02  IEB=IEB+NELMAT(ILB)
0209      02
0210      02  IF(INWATER.GT. 0. AND. INODE.EQ. NWATER+1) THEN
0211      03  WRITE(10, '( 3X,13,F10.3,B(1H-),1H+,1B(1H-))') INODE,ILB
0212      03  ELSE
0213      03  WRITE(10, '( 3X,13,F10.3,B(1H-),1H+, B(1H-))') INODE,'W'
0214      03  ENDIF
0215      02
0216      01  IF(INODE.LT. NNODE) THEN
0217      02  IF(INODE.GT. NWATER) THEN
0218      03  WRITE(10, '( 3X,3X, 10X,5X,14,12X,12))') INODE,ILB
0219      03  ELSE
0220      03  WRITE(10, '( 3X,3X, 10X,5X,14,12X,A2))') INODE,'W'
0221      03  ENDIF
0222      02
0223      01  1150 CONTINUE

```

```

0224, 01 C----- DO 3 I=1, NODE4 AT 2293
0225      01          DO 3 J=1, NODE4 AT 2303
0226, 01          3 AMM(I,J)=0, 0 AT 2313
0227, 02          DO 5 M=1, NELM AT 2325
0228          5 CALL EMMAT(M) AT 2334
0229, 01 C----- COMPUTE MATRICES [AJ, [BJ], [C] AT 2339
0230, 01          C----- DO 400 I=1, NODE4 AT 2349
0231, 01          DO 400 J=1, NODE4 AT 2357
0232          01          AMA(I,J)=0, 0 AT 2368
0233, 01          AMB(I,J)=0, 0 AT 2376
0234, 02          AMG(I,J)=0, 0 AT 2384
0235, 02          DO 500 M=1, NELM AT 2386
0236, 02          CALL EMMAT(M) AT 2395
0237, 02          AC0 CONTINUE
0238          02          DO 500 M=1, NELM
0239, 01          500 CALL EMMAT(M)
0240, 01 C----- COMPUTE MATRIX [C] (DAMPING) AT 2400
0241, 01          C----- DO 520 I=1, NODE4 AT 2410
0242          01          DO 520 J=1, NODE4 AT 2420
0243, 01          520 AC(I,J)=0, 0 AT 2432
0244, 02          DO 540 M=1, NELM AT 2441
0245          02          CALL ECMAT(M)
0246, 01          540 CALL ECMAT(M)
0247, 01 C----- Check the matrices
0248, 01          C----- CALL PRMAT(AMA, NODE4A, MSZ, 'MATRIX A ("h**2")')
0249, 01          CALL PRMAT(AMB, NODE4A, MSZ, 'MATRIX B ("ih")')
0250, 01          CALL PRMAT(AMG, NODE4A, MSZ, 'MATRIX C ("~1")')
0251, 01          CALL PRMAT(ACE, B, B, 'MATRIX D (damping)')
0252, 01          CALL PRMAT(AMM, 20, 40, 'MATRIX M (mass)')
0253, 01          C----- DO 10 I=1, NODE4A AT 2446
0254, 01          DO 10 J=1, NODE4A AT 2455
0255          01          10 AMT(I,J)=AMM(I,J)*1.E-2 AT 2464
0256, 01          DO 10 J=1, NODE4A AT 2478
0257, 02          10 AMT(I,J)=AMM(I,J)*1.E-2 AT 2478
0258          02          CALL TINVSS(NODE4A, AMT, DT, 1E-B, MSZ, IW, INDER)
0259          02          DO 15 J=1, NODE4A AT 2494
0260, 01          DO 15 J=1, NODE4A AT 2503
0261, 02          15 AMT(I,J)=AMT(I,J)*1.E-2 AT 2512
0262, 02 C----- set the value of wave number AT 2526
0263, 02          C----- 10=0 AT 2526
0264          02          IR=0 AT 2528
0265          02          DLTK=12./H/(6.*REAL(NAK1)/7.-1) AT 2529
0266          02          AK1=0, 0 AT 2547
0267          02          DO 1234 IAK1=1, NAK1 AT 2551
0268          01          IF(IAT .EQ. 0) THEN AT 2559
0269, 01          IF(IAK1 .GT. 1) THEN AT 2562
0270, 02          IF(IAK1*H .LT. 1.0) THEN AT 2567
0271, 03          AK1=AK1+DLTK/4. AT 2575
0272, 04          ELSE AT 2583
0273, 04          IF(IAK1*H .LT. 2.5) THEN AT 2584
0274, 04          AK1=AK1+DLTK/2. AT 2592
0275, 05          ELSE AT 2600
0276, 05          IF(IAK1*H .LT. 6.0) THEN AT 2601
0277, 05          AK1=AK1+DLTK AT 2609
0278, 06          ELSE AT 2615
0279, 06          AK1=AK1+1.5*DLTK AT 2616
0280, 06

```

```

0281.06          ENDIF
0282.05          ENDIF
0283.04          ENDIF
0284.03          ENDIF
0285.02          ELSE
0286.02          AK1=AKIDAT(IAK1)
0287.02          ENDIF
0288.01          WRITE(*,'(3X, ''hH='',F7.3,5X,1H(,12,1H))') AK1*h,IAK1,NAK1
0289.01          C
0290.01          NODEE=NODE4A
0291.01          construct the matrix for standard eigenvalue problem
0292.01          DO 40 I=1,NODEX
0293.02          DO 40 J=1,NODEX
0294.03          A0=AAR(1,J)=0.0
0295.01          DO 90 I=1,2*NODEX
0296.02          DO 90 J=1,2*NODEX
0297.03          90 AA1(I,J)=0.0
0298.01          DO 50 I=1,NODEX
0299.02          DO 50 J=1,NODEX
0300.03          IF(I .EQ. J) THEN
0301.04          AAR(1,J+NODEX)=1./FBAL
0302.04          ELSE
0303.04          AAR(1,J+NODEX)=0.0
0304.04          ENDIF
0305.03          50 CONTINUE
0306.01          DO 60 I=1,NODEX
0307.02          DO 60 J=1,NODEX
0308.03          AAR(1+NODEX,J)=0.0
0309.03          DO 60 K=1,NODEX
0310.04          60 AAR(1+NODEX,J)=AAR(1+NODEX,J)
0311.04          &           -DBLE( AMT(1,K)*(AMA(K,J)*AK1**2+AMG(K,J))*FBAL )
0312.01          DO 65 I=1,NODEX
0313.02          DO 65 J=1,NODEX
0314.03          AAI(1+NODEX,J)=0.0
0315.03          DO 65 K=1,NODEX
0316.04          65 AAI(1+NODEX,J)=AAI(1+NODEX,J)
0317.04          &           -DBLE( AMT(1,K)*AMB(K,J)*AK1*FBAL )
0318.01          DO 70 I=1,NODEX
0319.02          DO 70 J=1,NODEX
0320.03          AAR(1+NODEX,J+NODEX)=0.0
0321.03          DO 70 K=1,NODEX
0322.04          70 AAR(1+NODEX,J+NODEX)=AAR(1+NODEX,J+NODEX)
0323.04          &           -AMT(1,K)*AC(K,J)
0324.04          C
0325.04          C
0326.01          CALL CG(2*MST,2*NODE4A,AAR,AA1,WR,W1,1,ZR,Z1,FV1,FV2,FV3,IERR)
0327.01          DO 1100 J=1,NODE4A*2
0328.02          DO 1100 J=1,NODE4A*2
0329.03          1100 SL(J,I)=CMPLX(ZR(I,J),ZI(I,J))
0330.03          C
0331.01          WRITE(10,'(3X,''h='',F10.3)') AK1
0332.01          WRITE(10,'(3X,''hH='',F10.3)') AK1*h
0333.01          C
0334.01          WRITE(10,'(1/3X,''MODE'',5X,''1*DMEGA (Re)'',7x,''(Im)''
0335.01          &           ,6X,''omega1*H/V5'',4X,''-Re(1*omega)/omega1'')
0336.01          DO 880 I=NODE4A*2,1,-1
0337.02          1PASS(I)=

```

```

0338.02 IF(CABS(CMPLX(WR(1),WI(1)),WI(1)) .GT. 0.0 ) THEN
0339.03 ANDA=CABS(CMPLX(WR(1),WI(1))*H/VSA
0340.03 DMPA=-WR(1)/CABS(CMPLX(WR(1),WI(1)))
0341.03 IF(ANDA .LE. ANDMX .AND. WI(1) .GT. 0.00005 ) THEN
0342.04 JB1=4*(INWATER+1)-3
0343.04 JB3=4*(INWATER+1)-1
0344.04 IF(INWATER .GT. 0) THEN
0345.05 RCRT=0.0
0346.05 DO 6BB INWATER=1,INWATER
0347.06 JU1=4*INWATER-3
0348.06 JU3=4*INWATER-1
0349.06 IF(CABS(SL(1,JB1)) .GT. 0.0)
0350.06 & RCRT=RCRT+CABS(SL(1,JU1))/SL(1,JB1))
0351.06 IF(CABS(SL(1,JB3)) .GT. 0.0)
0352.06 & RCRT=RCRT+CABS(SL(1,JU3))/SL(1,JB3))
0353.06 CONTINUE
0354.05 RCRT=RCRT/REAL(INWATER)
0355.05 ELSE
0356.05 RCRT=0.0
0357.05 ENDIF
0358.04 IF(RCRT .LE. 1000.0 ) THEN
0359.05 IPASS(1)=*,*
0360.05 IO=10+1
0361.05 WNG(10)=AK1*H
0362.05 ANDR(10)=-WR(1)*H/VSA
0363.05 ANDI(10)=WI(1)*H/VSA
0364.05 ENDIF
0365.04 ENDIF
0366.03 WRITE(10,'(H,A1,I3,5X,F13.4,2X,F13.4,2X,F13.4,4)')
0367.03 & IPASS(1),NODE4A*2-I+1,WR(1),WI(1),ANDA,DMPA
0368.03 ELSE
0369.03 WRITE(10,'(H,A1,I3,5X,F13.4,2X,F13.4)')
0370.03 & IPASS(1),NODE4A*2-I+1,WR(1),WI(1)
0371.03 ENDIF
0372.02 880 CONTINUE
0373.01 NR=IR
0374.01 C
0375.01 JB1=4*(INWATER+1)-3
0376.01 JB2=4*(INWATER+1)-2
0377.01 JB3=4*(INWATER+1)-1
0378.01 JB4=4*(INWATER+1)
0379.01 C
0380.01 DO 80 I=NODE4A*2, 1 , -1
0381.02 IF(IPASS(1) .EQ. 0 ) GO TO 80
0382.02 WRITE(10,'(//,MODE= '//,15//)') NODE4A*2-I+1
0383.02 C
0384.02 IF(CABS(SL(1,JB1)) .GT. CABS(SL(1,JB3)) ) THEN
0385.02 C
0386.03 SLS1=1.0
0387.03 SLS2=SL(1,JB2)/SL(1,JB1)
0388.03 SLS3=SL(1,JB3)/SL(1,JB1)
0389.03 SLS4=SL(1,JB4)/SL(1,JB1)
0390.03 IR=IR+1
0391.03 IC(IR)=1
0392.03 WNG2(IR)=AK1*H
0393.03 IF(AIMAG(SLS3) .GT. 0.0 ) THEN
0394.04 UOW(IR)=-CABS(SL,S3)

```

```

0395.04          ELSE          AT 4240
0396.04          UDW(IR)= CAB(SL53)   AT 4241
0397.04          ENDIF         AT 4249
0398.03          ELSE          AT 4250
0399.03          IF(CABS( SL(1,JB3) ) .GT. 0.0) THEN   AT 4269
0400.03          C             SL51=SL(1,JB1)/SL(1,JB3)   AT 4278
0401.04          SL52=1.0      AT 4279
0402.04          SL52=SL(1,JB2)/SL(1,JB3)   AT 4302
0403.04          SL53=SL(1,JB4)/SL(1,JB3)   AT 4306
0404.04          IR=IR+1     AT 4340
0405.04          IC(IR)=2     AT 4374
0406.04          WNG2(IR)=AK1*H   AT 4376
0407.04          IF( AIMAG(SL51) .LT. 0.0) THEN   AT 4380
0408.04          UDW(IR)=-CABS(SL51)   AT 4388
0409.05          ELSE          AT 4392
0410.05          UDW(IR)= CAB(SL51)   AT 4401
0411.05          ENDIF         AT 4402
0412.05          ELSE          AT 4410
0413.04          IF(CABS( SL(1,JB2) ) .GT. 0.0) THEN   AT 4411
0414.04          SL51=0.0      AT 4431
0415.05          SL52=0.0      AT 4435
0416.05          SL53=0.0      AT 4439
0417.05          SL52=1.0      AT 4443
0418.05          SL54=SL(1,JB4)/SL(1,JB2)   AT 4478
0419.05          ELSE          AT 4479
0420.05          IF(CABS( SL(1,JB4) ) .GT. 0.0) THEN   AT 4499
0421.06          SL51=0.0      AT 4503
0422.06          SL52=0.0      AT 4507
0423.06          SL53=0.0      AT 4511
0424.06          SL54=1.0      AT 4515
0425.06          ELSE          AT 4516
0426.06          SL51=0.0      AT 4520
0427.06          SL52=0.0      AT 4524
0428.06          SL53=0.0      AT 4528
0429.06          SL54=0.0      AT 4532
0430.06          ENDIF         AT 4532
0431.05          ENDIF         AT 4532
0432.04          ENDIF         AT 4532
0433.03          C             DO 72 J=1,NNODE-1   AT 4532
0434.03          02          IF(J .EQ. NMATER+1) GO TO 72   AT 4543
0435.02          J1=4*j-3     AT 4549
0436.03          J2=4*j-2     AT 4554
0437.03          J3=4*j-1     AT 4559
0438.03          J4=4*j      AT 4564
0439.03          IF(CABS( SL(1,JB1) ) .GT. 0.0) SL(1,J1)=SL(1,JB1)   AT 4568
0440.03          IF(CABS( SL(1,JB2) ) .GT. 0.0) SL(1,J2)=SL(1,JB2)   AT 4631
0441.03          IF(CABS( SL(1,JB3) ) .GT. 0.0) SL(1,J3)=SL(1,JB3)   AT 4696
0442.03          IF(CABS( SL(1,JB4) ) .GT. 0.0) SL(1,J4)=SL(1,JB4)   AT 4759
0443.03          72 CONTINUE   AT 4824
0444.03          IF(CABS( SL(1,JB1) ) .GT. 0.0) SL(1,JB1)=1.0   AT 4825
0445.03          IF(CABS( SL(1,JB2) ) .GT. 0.0) SL(1,JB2)=1.0   AT 4861
0446.02          IF(CABS( SL(1,JB3) ) .GT. 0.0) SL(1,JB3)=1.0   AT 4899
0447.02          IF(CABS( SL(1,JB4) ) .GT. 0.0) SL(1,JB4)=1.0   AT 4935
0448.02          ENDIF         AT 4973
0449.02          IF(CABS( SL(1,JB1) ) .GT. 0.0) SL(1,JB1)=1.0
0450.02          IF(CABS( SL(1,JB2) ) .GT. 0.0) SL(1,JB2)=1.0
0451.02          IF(CABS( SL(1,JB3) ) .GT. 0.0) SL(1,JB3)=1.0
0452.02          IF(CABS( SL(1,JB4) ) .GT. 0.0) SL(1,JB4)=1.0
0453.02          WRITE(10,655)

```

```

0452.02      655 FORMAT(1X,4HNODE,12X,'Ux(r)',9X,'Ux(th)',8X,'Uy(r)',9X,'Uy(th)',  

0453.02      & 10X,'Wx(r)',9X,'Wx(th)',8X,'Wy(r)',9X,'Wy(th)',)  

0454.02      C DO 82 J=1,NODE-1  

0455.02      J1=A*J-3  

0456.03      J2=A*J-2  

0457.03      J3=A*J-1  

0458.03      J4=A*J  

0459.03      IF(CABS(SL(1,J1)), GT, 0.0) THEN  

0460.03      THSL(1)=ATAN2( AIMAG(SL(1,J1)), REAL(SL(1,J1)) )*180. /PI  

0461.04      ELSE  

0462.04      THSL(1)=0.0  

0463.04      ENDIF  

0464.04      IF(CABS(SL(1,J2)), GT, 0.0) THEN  

0465.03      THSL(2)=ATAN2( AIMAG(SL(1,J2)), REAL(SL(1,J2)) )*180. /PI  

0466.04      ELSE  

0467.04      THSL(2)=0.0  

0468.04      ENDIF  

0469.04      IF(CABS(SL(1,J3)), GT, 0.0) THEN  

0470.03      THSL(3)=ATAN2( AIMAG(SL(1,J3)), REAL(SL(1,J3)) )*180. /PI  

0471.04      ELSE  

0472.04      THSL(3)=0.0  

0473.04      ENDIF  

0474.04      IF(CABS(SL(1,J4)), GT, 0.0) THEN  

0475.03      THSL(4)=ATAN2( AIMAG(SL(1,J4)), REAL(SL(1,J4)) )*180. /PI  

0476.04      ELSE  

0477.04      THSL(4)=0.0  

0478.04      ENDIF  

0479.04      WRITE(10,660) J, CABS(SL(1,J1)), THSL(1), CABS(SL(1,J3)), THSL(3)  

0480.03      & , CABS(SL(1,J2)), THSL(2), CABS(SL(1,J4)), THSL(4)  

0481.03      660 FORMAT(1H , 'J= ', I3, 3X, 4(2F13.3, 3X))  

0482.03      82 CONTINUE  

0483.03      C  

0484.03      IF(CABS(SL51), GT, 0.0) THEN  

0485.02      THSL(1)=ATAN2( AIMAG(SL51), REAL(SL51) )*180. /PI  

0486.03      ELSE  

0487.03      THSL(1)=0.0  

0488.03      ENDIF  

0489.03      IF(CABS(SL52), GT, 0.0) THEN  

0490.02      THSL(2)=ATAN2( AIMAG(SL52), REAL(SL52) )*180. /PI  

0491.03      ELSE  

0492.03      THSL(2)=0.0  

0493.03      ENDIF  

0494.03      IF(CABS(SL53), GT, 0.0) THEN  

0495.02      THSL(3)=ATAN2( AIMAG(SL53), REAL(SL53) )*180. /PI  

0496.03      ELSE  

0497.03      THSL(3)=0.0  

0498.03      ENDIF  

0499.03      IF(CABS(SL54), GT, 0.0) THEN  

0500.02      THSL(4)=ATAN2( AIMAG(SL54), REAL(SL54) )*180. /PI  

0501.03      ELSE  

0502.03      THSL(4)=0.0  

0503.03      ENDIF  

0504.03      WRITE(10, '(/2X, ''/netin'', 2X,A(2F13.4,3X))')  

0505.02      & CABS(SL51), THSL(1), CABS(SL53), THSL(3)  

0506.02      & , CABS(SL52), THSL(2), CABS(SL54), THSL(4)  

0507.02      &  

0508.02      C

```

```

0509.02      80 CONTINUE
0510.01      1234 CONTINUE
0511          CLOSE(10, STATUS='KEEP')
0512          NG=10
0513          NR=IR
C-----OPEN(11, FILE=FL11, FORM='UNFORMATTED')
0514          AT 5767
0515          AT 5768
0516          AT 5769
0517          AT 5785
0518          AT 5788
0519          AT 5791
0520          AT 5793
0521          AT 5813
0522          AT 5833
0523          AT 5870
0524          AT 5907
0525          AT 5944
0526          AT 5964
0527          AT 6001
0528          AT 6038
0529          AT 6074
C-----CLOSE(11, STATUS='KEEP')
0530          AT 6090
0531          AT 6137
0532          AT 6182
0533          AT 6188
C-----STOP
END

C*****SUBROUTINE EMMAT(M)
C-----0532
0533          SUBROUTINE EMMAT(M)
C-----0534          PARAMETER (MEL=17, MSZ=4*MEL)
0535          COMMON /ABGE/ AMAE(B,B), AMBE(B,B), AMME(B,B)
0536          COMMON /ABG/  AMA(MSZ,MSZ), AMB(MSZ,MSZ), AMG(MSZ,MSZ)
0537          DIMENSION LL(18)
C-----0538
0539          LL(1)=4*M-3
0540          LL(2)=4*M-2
0541          LL(3)=4*M-1
0542          LL(4)=4*M
0543          LL(5)=4*(M+1)-3
0544          LL(6)=4*(M+1)-2
0545          LL(7)=4*(M+1)-1
0546          LL(8)=4*(M+1)
0547          CALL KMAT(M)
0548          DD 10 I=1,B
0549          DO 10 J=1,B
0550          IT=LL(1)
0551          JT=LL(2)
0552          JT=LL(3)
0553          JT=LL(4)
0554          JT=LL(5)
0555          JT=LL(6)
0556          JT=LL(7)
0557          JT=LL(8)
C-----10 CONTINUE
          RETURN
END

C*****SUBROUTINE EMMAT(M)
C-----0558
0559          SUBROUTINE EMMAT(M)
C-----0550
0551
0552
0553
0554
0555
0556
0557

```

```

0560      C-----PARAMETER (MEL=17, MSZ=4*MEL)
0561      COMMON /ABGE/AMAE(B,B), AMBE(B,B), AMGE(B,B), AMME(B,B)
0562      COMMON /AMM/ AMM(MSZ,MSZ)
0563      DIMENSION LL(B)
0564
0565      C-----LL(1)=4*M-3
0566          LL(2)=4*M-2
0567          LL(3)=4*M-1
0568          LL(4)=4*M
0569          LL(5)=4*(M+1)-3
0570          LL(6)=4*(M+1)-2
0571          LL(7)=4*(M+1)-1
0572          LL(8)=4*(M+1)
0573          CALL CMAT(M)
0574          DO 10 I=1,B
0575          DO 10 J=1,B
0576    01   IT=LL(I)
0577    02   JT=LL(J)
0578          10 AMM(IT,JT)=AMM(IT,JT)+AMME(I,J)
0579    02
0580          RETURN
0581
0582      C*****SUBROUTINE ECMAT(M)
0583
0584      C-----PARAMETER (MEL=17, MSZ=4*MEL)
0585      COMMON /AC/AC(MSZ,MSZ)
0586      COMMON /ACE/ACE(B,B)
0587      COMMON /ACE/ACE(B,B)
0588      DIMENSION LL(B)
0589
0590      C-----LL(1)=4*M-3
0591          LL(2)=4*M-2
0592          LL(3)=4*M-1
0593          LL(4)=4*M
0594          LL(5)=4*(M+1)-3
0595          LL(6)=4*(M+1)-2
0596          LL(7)=4*(M+1)-1
0597          LL(8)=4*(M+1)
0598          CALL CMAT(M)
0599          DO 10 I=1,B
0600          DO 10 J=1,B
0601    01   IT=LL(I)
0602    02   JT=LL(J)
0603    02   10 AC(IT,JT)=AC(IT,JT)+ACE(I,J)
0604          RETURN
0605
0606      C*****SUBROUTINE KMAT(M)
0607
0608      C-----STIFFNESS MATRIX
0609      C-----PARAMETER (MEL=17, MPR=5)
0610

```

```

COMMON /ABGE/AMAE(B,B),AMBE(B,B),AMGE(B,B),AMME(B,B)
COMMON /GM/GM(MPR),PERM(MPR),PO(MPR),ALFA(MPR),ANR(MPR),RM(MPR)
      ,G(MPR),AKF,AKS(MPR),NWATER,LMAT(ML),NMAT
COMMON /HT/AK1,B(MEL),H,NELMAT(MPR),THCK(MPR)
DIMENSION DMAT(4,4), AMAT(4,4), ADA(4,4), ADB(4,4)
      ,BDA(4,4), BDB(4,4)

C-----IF(M . I.E. NWATER) THEN
      AT 7
      RMT=0.0
      GNT=0.0
      ALFT=1.0
      GT=AMF
      ELSE
      RMT=RM(LMAT(M))
      GNT=G(LMAT(M))
      ALFT=ALFA(LMAT(M))
      GT=G(LMAT(M))
      ENDIF
      DMAT(1,1)= RMT + 2.*GNT + ALFT**2*GT
      DMAT(1,2)= RMT + ALFT**2*GT
      DMAT(1,3)=0.
      DMAT(1,4)= ALFT*GT
      DMAT(2,2)= RMT + 2.*GNT + ALFT**2*GT
      DMAT(2,3)=0.
      DMAT(2,4)= ALFT*GT
      DMAT(3,3)= GNT
      DMAT(3,4)=0.
      DMAT(4,4)= GT
      DO 500 I=2,4
      DO 500 J=1, I-1
      500 DMAT(I,J)=DMAT(J,I)
      C-----DO 510 I=1,4
      DO 510 J=1,4
      510 AMAT(I,J)=0.0
      AMAT(1,1)=1.
      AMAT(4,2)=1.
      AMAT(3,3)=1.
      C-----CALL TMULT(AMAT,DMAT,AMAT,ADA)
      DO 640 I=1,4
      DO 640 J=1,4
      640 AMAE(I,J)=B(M)/3.*ADA(I,J)
      AMAE(I+4,J+4)=B(M)/3.*ADA(I,J)
      AMAE(I+4,J)=B(M)/6.*ADA(I,J)
      AMAE(I+4,J+4)=B(M)/6.*ADA(I,J)
      640 CONTINUE
      C-----CALL TMULT(BMAT,DMAT,BMAT,BDB)
      DO 740 I=1,4
      740

```



```

0722      AMME(1, 1)=W1          AT 115
0723      AMME(1, 2)=W2          AT 119
0724      AMME(1, 5)=V1          AT 123
0725      AMME(1, 6)=V2          AT 127
0726      AMME(2, 2)=W3          AT 131
0727      AMME(2, 5)=V2          AT 135
0728      AMME(2, 6)=V3          AT 139
0729      AMME(3, 3)=W1          AT 143
0730      AMME(3, 4)=W2          AT 147
0731      AMME(3, 7)=V1          AT 151
0732      AMME(3, 8)=V2          AT 155
0733      AMME(4, 4)=W3          AT 159
0734      AMME(4, 7)=V2          AT 163
0735      AMME(4, 8)=V3          AT 167
0736      AMME(5, 5)=W1          AT 171
0737      AMME(5, 6)=W2          AT 175
0738      AMME(6, 6)=W3          AT 179
0739      AMME(7, 7)=W1          AT 183
0740      AMME(7, 8)=W2          AT 187
0741      AMME(8, 8)=W3          AT 191
0742      DO 20 K=1, B           AT 195
0743      01                   AT 202
0744      02                   AT 210
0745      02                   AT 223
0746      20 CONTINUE          AT 225
0747      RETURN               AT 226
END

```

```

0748 C*****SUBROUTINE CMAT(M)*****
0749 C-----DAMPING MATRIX . . .
0750 C-----PARAMETER (MEL=17, MPR=5)
0751 C-----COMMON /ACE/ACE(B, B)
0752 C-----COMMON /GM/GM(MPR), PERM(MPR), PO(MPR), ALFA(MPR), ANR(MPR), RM(MPR)
0753 C-----COMMON /HT/HT(MEL), NMAT(MEL), NMAT
0754 C-----COMMON /KNS/KNS(MPR), NWATER, LMAT(MEL), NMAT
0755 C-----COMMON /HT/AK1,B1(MEL), H, NELMAT(MPR), THCK(MPR)
0756 C-----IF(M .LE. NWATER) THEN
0757 C-----  B1=0.0
0758      IF(M .LE. NWATER) THEN          AT 7
0759      01                   AT 11
0760      01                   AT 13
0761      01                   AT 14
0762      01                   AT 19
0763      02                   AT 65
0764      02                   AT 69
0765      02                   AT 76
0766      02                   AT 83
0767      02                   AT 94
0768      02                   AT 98
0769      01                   AT 102
0770      DO 10 J=1, B           AT 69
0771      01                   AT 76
0772      02                   AT 83
0773      10 ACE(1, J)=0.0          AT 94
0774      ACE(2, 2)=B2            AT 98
0775      ACE(2, 6)=B1            AT 102
0776      ACE(4, 4)=B2            AT 102

```

```

      ACE(4, B)=B1
      ACE(6, B)=B2
      ACE(8, B)=B2
      DD=20 K=1, B
      DD 20 J=N, B
      20 ACE(J, K)=ACE(K, J)
      RETURN
      END

C*****SUBROUTINE TINVSS(N, A, DD, E, NN, IW, INDER)
C-----C
      DIMENSION A(NN, N), IW(2*N)
      INDER=0
      IF(N=1) 910, 930, 101
      101 IF(N, GT, NN) GO TO 900
      EPS=0.0
      DD=1.0
      DD 100 K=1, N
      PIV=0.0
      IF(PIV, EQ, 0) EPS=ABS(PIV)*E
      IF(PIV, EQ, K) GO TO 130
      DD=-DD
      DD 120, J=1, N
      WORK=A(JPIV, J)
      A(IPIV, J)=A(K, J)
      120 A(K, J)=WORK
      130 IF(JPIV, EQ, K) GO TO 150
      DD=-DD
      DD 140 I=1, N
      WORK=A(I, JPIV)
      A(I, JPIV)=A(I, K)
      140 A(K, J)=WORK
      150 IW(2*N-1)=IPIV
      AA=1.0/PIV
      IW(2*N)=JPIV
      DD 210 J=1, N
      210 A(K, J)=A(K, J)*AA
      DD 220 I=1, N
      IF(I, EQ, K) GO TO 220
      AZ=A(I, K)
      IF(ABS(AZ), LE, 0.0) GO TO 220
      220 A(I, J)=A(I, J)-AZ*A(Z)
      A(I, K)=AZ*A(Z)
      230 A(I, J)=A(I, J)-A(K, J)*AZ
      AT 106
      AT 110
      AT 114
      AT 116
      AT 123
      AT 131
      AT 147
      AT 148
      AT 24
      AT 31
      AT 37
      AT 41
      AT 45
      AT 55
      AT 59
      AT 71
      AT 83
      AT 105
      AT 107
      AT 109
      AT 120
      AT 122
      AT 128
      AT 137
      AT 149
      AT 153
      AT 158
      AT 168
      AT 179
      AT 196
      AT 208
      AT 212
      AT 217
      AT 227
      AT 238
      AT 256
      AT 268
      AT 275
      AT 281
      AT 288
      AT 298
      AT 312
      AT 322
      AT 325
      AT 336
      AT 340
      AT 350
      AT 374
      MATRIX INVERSION . . .

```

```

0830.02      220 CONTINUE
0831.01      100 A(I,K)=AA
0832.01      DD 400 KK=2, N
          K=N+1-KK
          IJ=IW(2*KK)
          IF (IJ EQ K) GO TO 420
          DO 410 J=1, N
          WORK=A(I,J,J)
          A(IJ,J)=A(K,J)
          410 A(K,J)=WORK
          420 IJ=IW(2*KK-1)
          IF (IJ EQ K) GO TO 400
          DO 430 I=1, N
          WORK=A(I,IJ)
          A(I,IJ)=A(I,K)
          430 A(I,K)=WORK
          400 CONTINUE
          RETURN
0847.01      910 INDER=-1
0848.02      WRITE(6,691) N
0849.02      691 FORMAT(1H , '*** N = ', 15/)
0850.02      RETURN
0851.01      900 INDER=-1
0852.02      WRITE(6,691) N,NN
0853.02      691 FORMAT(1H , '*** N= ', 15, 5X, ' *** NN= ', 15/)
0854.02      RETURN
0855.02      690 FORMAT(1H , '*** N= ', 15, 5X, ' *** NN= ', 15/)
0856.02      920 DD=0. 0
          INDER=N-K+1
0857.02      NNN=K-1
0858.02      WRITE(6,692) NNN
0859.02      692 FORMAT(1H , '*** NNN = ', 15/)
0860.02      RETURN
0861.01      930 DD=A(1,1)
0862.02      K=1
          IF (ABS(DD).LE.0.0) GO TO 920
          A(1,1)=1. 0/A(1,1)
          RETURN
0866.02      END
0867.02

```

---

```

0868.02      C*****SUBROUTINE TMULT(AMAT,DMAT,BMAT,ADB)
0869.02      C-----DIMENSION AMAT(4,4),DMAT(4,4),BMAT(4,4),TEMP(4,4)
0870.02      C-----0871.02      DD 700 I=1,4
0872.02      C-----0873.01      DD 700 J=1,4
0874.02      C-----0875.02      DD 710 K=1,4
0876.03      C-----0877.02      710 SS=SS+AMAT(K,1)*DMAT(K,J)
0878.02      C-----0879.01      TEMP(I,J)=SS
          700 CONTINUE
          DD 720 I=1,4
          0880.01      DD 720 J=1,4
          0881.02      SS=0. 0
          0882.02      DD 730 K=1,4
          0883.03      730 SS=SS+TEMP(I,K)*BMAT(K,J)

```

```

0884 02      ADB(1,J)=SS          AT 103
0885 02      720 CONTINUE        AT 113
0886          RETURN            AT 115
0887          END               AT 116

0888          C*****SUBROUTINE PRMAT(A,NJ,MJ,K,MMSH)*****
0889          C-----COMMON /LPO/1LPDOUT,1PLDTR
0890          C-----DIMENSION A(1)
0891          C-----CHARACTER MMSH*(*),MSH1*66,MSH2*70,MFORM*30
0892          C-----IF(1LPDOUT .NE. 0) NCOL=10          AT 11
0893          C-----IF(1LPDOUT .EQ. 0) NCOL=6
0894          C-----NPAGE=MJ/NCOL+MIN(1,MOD(MJ,NCOL))          AT 17
0895          C-----MSH1=MMSH//,'/'          AT 34
0896          C-----MSH2=' //,'/MSH1          AT 56
0897          C-----DO 100 IPAGE=1,NPAGE          AT 77
0898          C-----JFR=(IPAGE-1)*NCOL+1          AT 86
0899          C-----JTO=MIN(IPAGE*NCOL,MJ)          AT 91
0900          C-----WRITE(MFORM,'(1I1H//A70//5X,,12,BH17,4X))')' NCOL
0901 01      JTO=1          AT 102
0902 01      WRITE(6,MFORM) MSH1(J,J=JFR,JTO)          AT 149
0903 01      WRITE(6,MFORM) '(4H15,,12,6HE11.4)' NCOL          AT 193
0904 01      WRITE(6,MFORM) MSH1(J,J=JFR,JTO)          AT 235
0905 01      WRITE(6,MFORM) '(4H15,,12,6HE11.4)' NCOL          AT 245
0906 01      DO 110 I=1,NJ          AT 301
0907 02      110 WRITE(6,MFORM) I, I*(J-1)*K+I), J=JFR, JTO          AT 302
0908 01      100 CONTINUE          AT 303
0909          RETURN
0910          END

0911          C*****SUBROUTINE RDDTITLE(NELM,ROWS,ROWF,NAK1,IAT,AK1DAT,OUTFL,FL11)*****
0912          C-----COMMON /GM/MPR,PERM(MPR),PO(MPR),ANR(MPR),RM(MPR)
0913          C-----PARAMETER (NEL=17,MPR=5)
0914          C-----COMMON /GM/MPR,PERM(MPR),PO(MPR),ANR(MPR),RM(MPR)
0915          C-----& COMMON /HT/AK1,BINEL,H,NELMAT(MPR),NMAT
0916          C-----CHARACTER TITLE(10)*4, OUTFL*30, FL11*30
0917          C-----DIMENSION ROWS(MPR),AK1DAT(100)          AT 52
0918          C-----READ(2,'(10A4)') TITLE(I), I=1,10          AT 89
0919          C-----READ(2,'(215)') NMAT,NWATER          AT 91
0920          C-----IELFR=1          AT 101
0921          C-----DO 110 IMAT=1,NMAT          AT 143
0922          C-----READ(2,'(215)') IMAT,NELMAT(IMAT)          AT 256
0923          C-----IELTO=IELFR+NELMAT(IMAT)-1          AT 262
0924          C-----DO 100 IELM=IELFR,IELTO          AT 272
0925 01      C-----B(IELM)=THCK(IMAT)/REAL(NELMAT(IMAT))          AT 28B
0926 01      C-----LMAT(IELM)=IMAT          AT 293
0927 01      C-----IELFR=IELT0+1          AT 294
0928 01      C-----100 CONTINUE          AT 297
0929 01      C-----IELFR=IELT0+1          AT 297
0930 02      C-----110 CONTINUE          AT 297
0931 02      C-----IELFR=IELT0+1          AT 297
0932 02      C-----100 CONTINUE          AT 297
0933 01      C-----IELFR=IELT0+1          AT 297
0934 01      C-----110 CONTINUE          AT 297

```

SALFORD UNIVERSITY FTN77 VER. 2304S

<ENG9>AKIRA>SEC>RLW.FTN77

Page 18

22:17:31

SUN, AUG 06 1989

```
0535      NELM=IELTO
0536      READ(2, '(2F10.0)') AKF, RDWF
0537      READ(2, '(2I5)') NAK1, IAT
0538      IF(IAT .NE. 0) READ(2, '(7F10.0)') (AKIDAT(j), j=1, NAK1)
0539      READ(2, '(A30)', END=200) OUTFL
0540      READ(2, '(A30)', END=210) FL11
0541      RETURN
0542      200 OUTFL='RLW OUT'
0543      210 FL11='RLW_RESULT. DATA'
0544      RETURN
0545      END
```

END OF COMPILATION CLOCKED 8.248 SECONDS

COMPILER OPTIONS: LISTING INTS NOMAP NOCHECK NOBIG LOGS DYNM OFFSET NOANSI NODEBUG NOPAGE THROW NDFRN  
FPN NOLUNFREC NO\_OPTIMISE NOIMPURE

0001 C\*\*\*\*\*  
 0002 SUBROUTINE CG(NM,N,AR,AI,WR,WI,MATZ,ZR,ZI,FV1,FV2,FV3,IERR)  
 0003 C---  
 0004 INTEGER NM,IS1,IS2,IERR,MATZ  
 0005 REAL#B AR(NM,N),AI(NM,N),WR(N),WI(N),ZR(NM,N),ZI(NM,N),  
 0006 \$ FV1(N),FV2(N),FV3(N)  
 0007 C---  
 0008 C This subroutine calls the recommended sequence of  
 0009 C subroutines from the eigensystem subroutine package (EISPACK)  
 0010 C to find the eigenvalues and eigenvectors (if desired)  
 0011 C of a complex general matrix.  
 0012 C---  
 0013 C  
 On input:  
 0014 C  
 0015 NM must be set to the row dimension of the two-dimensional  
 0016 array parameters as declared in the calling program  
 0017 dimension statement;  
 0018 C  
 0019 C N is the order of the matrix A=(AR,AI);  
 0020 C  
 0021 C AR and AI contain the real and imaginary parts,  
 0022 C respectively, of the complex general matrix;  
 0023 C  
 0024 C MATZ is an integer variable set equal to zero if  
 0025 C only eigenvalues are desired; otherwise it is set to  
 0026 C any non-zero integer for both eigenvalues and eigenvectors.  
 0027 C  
 On output:  
 0028 C  
 0029 C WR and WI contain the real and imaginary parts,  
 0030 C respectively, of the eigenvalues;  
 0031 C  
 0032 C ZR and ZI contain the real and imaginary parts,  
 0033 C respectively, of the eigenvectors if MATZ is not zero;  
 0034 C  
 0035 C IERR is an integer output variable set equal to an  
 0036 C error completion code described in section 2B of the  
 0037 C documentation. The normal completion code is zero;  
 0038 C  
 0039 C FV1, FV2, and FV3 are temporary storage arrays.  
 0040 C  
 0041 C  
 0042 C Questions and comments should be directed to B. S. Garbow,  
 0043 C Applied Mathematics division, Argonne National Laboratory  
 0044 C  
 0045 C  
 0046 C  
 0047 C IF (N .LE. NM) GO TO 10  
 0048 C IERR = 10 \* N  
 0049 C GO TO 50  
 0050 C  
 0051 C 10 CALL CBAL(NM,N,AR,AI,IS1,IS2,FV1) AT 25  
 0052 C AT 29  
 0053 C AT 30

```

0053      CALL CORTH(NM,N,IS1,IS2,AR,AI,FV2,FV3)          AT 46
0054      IF (MATZ .NE. 0) GO TO 20
0055      C       ::::: find eigenvalues only ::::::::::::
0056      CALL COMOR(NM,N,IS1,IS2,AR,AI,WR,WI,JERR)      AT 64
0057      GO TO 50                                         AT 68
0058      C       ::::: find both eigenvalues and eigenvectors ::::::::::::
0059      20 CALL COMOR2(NM,N,IS1,IS2,FV2,FV3,AR,AI,WR,WI,ZR,ZI,JERR)   AT 88
0060      IF (JERR .NE. 0) GO TO 50                         AT 89
0061      CALL CBABK2(NM,N,IS1,IS2,FV1,N,ZR,ZI)           AT 117
0062      50 RETURN                                         AT 121
0063      C       ::::: last card of CG ::::::::::::
0064      END                                              AT 140

```

```

0065 C*****SUBROUTINE CBABK2(NM,N,LOW,IGH,SCALE,M,ZR,ZI)
0066 C-----0067
0067      INTEGER I,J,K,M,N,II,NM,IGH,SCALE,M,ZR,ZI)
0068      REAL*B8 SCALE(NM),ZR(NM,M),ZI(NM,M)
0069      REAL*B8 S
0070
0071      C This subroutine is a translation of the Algol procedure
0072      C CBABK2, which is a complex version of BALBAK,
0073      C Num. Math. 13, 293-304(1969) by Parlett and Reinsch,
0074      C Handbook for auto. comp., Vol.II-Linear Algebra, 315-326(1971).
0075
0076      C This subroutine forms the eigenvectors of a complex general
0077      C matrix by back transforming those of the corresponding
0078      C balanced matrix determined by CBAL.
0079
0080      C On input:
0081
0082      C NM must be set to the raw dimension of two-dimensional
0083      C array parameters as declared in the calling program
0084      C DIMENSION statement;
0085
0086      C N is the order of the matrix;
0087
0088      C LOW and IGH are integers determined by CBAL;
0089
0090      C SCALE contains information determining the permutations
0091      C and scaling factors used by CBAL;
0092
0093      C M is the number of eigenvectors to be back transformed;
0094
0095      C ZR and ZI contain the real and imaginary parts,
0096      C respectively, of the eigenvectors to be
0097      C back transformed in their first M columns.
0098
0099      C On output:
0100
0101      C ZR and ZI contain the real and imaginary parts,
0102      C respectively, of the transformed eigenvectors
0103      C in their first M columns.
0104
0105      C Questions and comments should be directed to B. S. Garbow,
0106

```

Applied Mathematics Division, Argonne National Laboratory

```

0107 C
0108 C
0109 C
0110 C
0111 C IF (M .EQ. 0) GO TO 200
0112 C IF (IGH .EQ. LOW) GO TO 120
0113 C DO 110 I = LOW, IGH
0114 C S = SCALE(I)
0115 01 C :left hand eigenvectors are back transformed
0116 01 C if the foregoing statement is replaced by
0117 01 C S=1.0D0/SCALE(I).
0118 01 C DO 100 J = 1, M
0119 01 C ZR(I,J) = ZR(I,J) * S
0120 02 C ZI(I,J) = ZI(I,J) * S
0121 02 C 100 CONTINUE
0122 02 C
0123 02 C
0124 01 C 110 CONTINUE
0125 01 C :FOR I=LOW-1 STEP -1 UNTIL 1,
0126 01 C 120 DO 140 II = 1, N
0127 01 C :10H+1 STEP 1 UNTIL N DO --- :::::
0128 01 C I = II
0129 01 C IF (I .GE. LOW .AND. I .LE. 10H) GO TO 140
0130 01 C IF (I .LT. LOW) I = LOW - II
0131 01 C K = SCALE(I)
0132 01 C IF (K .EQ. 1) GO TO 140
0133 01 C
0134 01 C DD 130 J = 1, M
0135 02 C S = ZR(I,J)
0136 02 C ZR(I,J) = ZR(K,J)
0137 02 C ZR(K,J) = S
0138 02 C S = ZI(I,J)
0139 02 C ZI(I,J) = ZI(K,J)
0140 02 C ZI(K,J) = S
0141 02 C 130 CONTINUE
0142 02 C
0143 01 C 140 CONTINUE
0144 01 C
0145 200 RETURN
0146 C :last card of CBANK2 :::::::
0147 END

```

```

0148 C*****
0149 C SUBROUTINE CBAL(NM,N,AR,AI,LOW,IGH,SCALE)
0150 C

```

```

0151 INTEGER I,J,K,L,M,N,JI,NM,IGH,LOW,IEXC
0152 REAL*8 AR(NM,N),AI(NM,N),SCALE(N)
0153 REAL*8 C,F,G,R,S,B2,RADIX
0154 REAL*8 DABS
0155 LOGICAL NDCONDV
0156 C

```

```

0157 C
0158 C This subroutine is a translation of the ALGOL procedure
0159 C CBALANCE, which is a complex version of BALANCE
0160 C Num. Math. 13, 293-304(1969) by Parlett and Reinsch.

```

C Handbook for Auto. Comp., Vol. III-Linear Algebra, 315-326(1971).

C This subroutine balances a complex matrix and isolates  
C eigenvalues whenever possible.

C On input:

0161 C NM must be set to the row dimension of two-dimensional  
0162 C array parameters as declared in the calling program  
0163 C DIMENSION statement;

0164 C N is the order of the matrix;

0165 C AR and AI contain the real and imaginary parts,  
0166 C respectively, of the complex matrix to be balanced.

C On output:

0167 C AR and AI contain the real and imaginary parts,  
0168 C respectively, of the balanced matrix;  
0169 C LLOW and IGH are two integers such that AR(I,J) and AI(I,J)  
0170 C are equal to zero if  
0171 C (1) J is greater than J and  
0172 C (2) J=1,...,LLW-1 or I=IGH+1,...,N;

0173 C SCALE contains information determining the  
0174 C permutations and scaling factors used.  
0175 C Suppose that the principal submatrix in rows LLOW through IGH  
0176 C has been balanced, that P(J) denotes the index interchanged  
0177 C with J during the permutation step, and that the elements  
0178 C of the diagonal matrix used are denoted by D(I,J). Then  
0179 C  
0180 C      SCALE(J) = P(J),      FOR J = 1,...,LLW-1  
0181 C                            J = LLOW,...,IGH  
0182 C                            I = IGH+1,...,N.  
0183 C The order in which the interchanges are made is N to IGH+1,  
0184 C then 1 to LLW-1.

0185 C Note that J is returned for IGH if IGH is zero formally.  
0186 C  
0187 C The ALGOL procedure EXC contained in cbalance appears in  
0188 C CRAL in line. (Note that the ALGOL roles of identifiers  
0189 C K,L have been reversed.)  
0190 C Arithmetic is real throughout.  
0191 C Questions and comments should be directed to B. S. Garbow,  
0192 C Applied Mathematics Division, Argonne National Laboratory  
0193 C

0194 C  
0195 C  
0196 C  
0197 C  
0198 C  
0199 C  
0200 C  
0201 C  
0202 C  
0203 C  
0204 C  
0205 C  
0206 C  
0207 C  
0208 C  
0209 C  
0210 C  
0211 C  
0212 C  
0213 C  
0214 C  
0215 C  
0216 C  
0217 C

..... RADIX is a machine dependent parameter specifying  
..... the base of the machine floating point representation.  
RADIX = 16.0D0 for long form arithmetic  
on S360 : : : : :  
DATA RADIX/2/

```

0218      C     B2 = RADIX * RADIX          AT 18
0219      C     K = 1                      AT 24
0220      C     L = N                      AT 26
0221      C     GO TO 100                 AT 29
0222      C     ..... In-line procedure for row and
0223      C     column exchange ::::::::::::
0224      C     20 SCALE(M) = J
0225      C     IF (J .EQ. M) GO TO 50
0226      C
0227      C     DO 30 I = 1, L
0228      C     F = AR(I,J)
0229      C     AR(I,J) = AR(I,M)
0230      C     AR(I,M) = F
0231      C     AI(I,M) = F
0232      C     F = AI(I,J)
0233      C     AI(I,J) = AI(I,M)
0234      C     AI(I,M) = F
0235      C     30 CONTINUE
0236      C
0237      C     DD 40 I = K, N          AT 133
0238      C     F = AR(J,I)
0239      C     AR(J,I) = AR(M,I)
0240      C     AR(M,I) = F
0241      C     F = AI(J,I)
0242      C     AI(J,I) = AI(M,I)
0243      C     AI(M,I) = F
0244      C     40 CONTINUE
0245      C
0246      C     50 GO TO (B0,130), TEXP          AT 224
0247      C     ..... Search for rows isolating an eigenvalue
0248      C     and push them down ::::::::::::
0249      C     80 IF (L .EQ. 1) GO TO 280          AT 229
0250      C     L = L - 1
0251      C     ..... FOR J=L STEP -1 UNTIL 1 DO -- ::::::::::::
0252      C     100 DO 120 JJ = 1, L          AT 233
0253      C     J = L + 1 - JJ
0254      C
0255      C     DO 110 I = 1, L
0256      C     IF (I .EQ. J) GO TO 110          AT 236
0257      C     IF (AR(I,I) .NE. 0.0D0 .OR. AI(I,I) .NE. 0.0D0) GO TO 120          AT 245
0258      C     120 CONTINUE
0259      C
0260      C     M = L
0261      C     TEXP = 1
0262      C     GO TO 20
0263      C     120 CONTINUE
0264      C
0265      C     GO TO 140
0266      C     ..... Search for columns isolating an eigenvalue
0267      C     and push them left ::::::::::::
0268      C     130 K = K + 1
0269      C
0270      C     140 DO 170 J = K, L
0271      C
0272      C     DO 150 I = K, L
0273      C     IF (I .EQ. J) GO TO 150          AT 287
0274      C     IF (AR(I,I) .NE. 0.0D0 .OR. AI(I,I) .NE. 0.0D0) GO TO 170          AT 288
0275      C

```

```

0275.02   150    CONTINUE
0276.02   C      M = K
0277.01   C      JEXC = 2
0278.01   C      GO TO 20
0279.01   C      170    CONTINUE
0280.01   C      ::::::: New balance the submatrix in rows K to L :::::::
0281.01   C      DO 180 I = K, L
0282.01   C      180    SCALE(I) = 1.0D0
0283.01   C      ::::::: iterative loop for norm reduction :::::::
0284.01   C      190    NDCONV = .FALSE.
0285.01   C
0286.01   C
0287.01   C      DD 270 I = K, L
0288.01   C      C = 0.0D0
0289.01   C      R = 0.0D0
0290.01   C
0291.01   C      DD 200 J = K, L
0292.02   C      IF ((J .EQ. 1) GO TO 200
0293.02   C      C = C + DABS(AR(I,J)) + DABS(AI(I,J))
0294.02   C      R = R + DABS(AR(I,J)) + DABS(AI(I,J))
0295.02   C      200    CONTINUE
0296.02   C      ::::::: guard against zero C or R due to underflow :::::::
0297.01   C      IF ((C .EQ. 0.0D0 .OR. R .EQ. 0.0D0) GO TO 270
0298.01   C      G = R / RADIX
0299.01   C      F = 1.0D0
0300.01   C      S = C + R
0301.01   C      210    IF ((C .GE. 0) GO TO 220
0302.01   C      F = F * RADIX
0303.01   C      C = C * B2
0304.01   C      GO TO 210
0305.01   C      220    G = R * RADIX
0306.01   C      230    IF ((C .LT. 0) GO TO 240
0307.01   C      F = F / RADIX
0308.01   C      C = C / B2
0309.01   C      GO TO 230
0310.01   C      ::::::: New balance :::::::
0311.01   C      240    IF (((C + R) / F .GE. 0.95D0 * S) GO TO 270
0312.01   C      G = 1.0D0 / F
0313.01   C      SCALE(I) = SCALE(I) * F
0314.01   C      NDCONV = .TRUE.
0315.01   C
0316.01   C      DO 250 J = K, N
0317.02   C      AR(I,J) = AR(I,J) * G
0318.02   C      AI(I,J) = AI(I,J) * G
0319.02   C      250    CONTINUE
0320.02   C
0321.01   C      DO 260 J = 1, L
0322.02   C      AR(J,I) = AR(J,I) * F
0323.02   C      AI(J,I) = AI(J,I) * F
0324.02   C      260    CONTINUE
0325.02   C
0326.01   C      270    CONTINUE
0327.01   C
0328.01   C      IF (NDCONV) GO TO 190
0329.01   C      280    LOW = K
0330.01   C      IGH = L

```

```

0332      RETURN          AT 637
0333      C      ::::::: last card of CBAL :::::::
0334      END             AT 638

0335      C*****SUBROUTINE COMQR(NM,N,LOW,IGH,HR,HI,WR,WI,IERR)
0336      C-----REAL#B DREAL,DIMAG
0337      C-----INTEGER I,J,L,N,ENLL,NM,IGH,ITS,LOW,LP1,ENM1,IERR
0338      C-----REAL#B HR(NM,N),HI(NM,N),WR(N),WI(N)
0339      C-----REAL#B SI,SR,TI,TR,XI,XR,YI,YR,ZI,ZR,NORM,MACHEP
0340      C-----REAL#B DSORT,CDABS,DAB5
0341      C-----INTEGER MINO
0342      C-----COMPLEX#16 Z3
0343      C-----COMPLEX#16 CDSORT,DCMPLX
0344      C-----REAL#B DREAL,DIMAG
0345      C-----Statement functions, enable extraction of real and
0346      C-----imaginary parts of double precision complex numbers :::::::
0347      C-----DREAL(Z3) = Z3
0348      C-----DIMAG(Z3) = (0.0D0,-1.0D0)*Z3          AT 18
0349      C-----DREAL(-1.0D0) = 0.0D0
0350      C-----DIMAG(-1.0D0) = -1.0D0
0351      C-----This subroutine is a translation of a unitary analogue of the
0352      C-----ALGOL procedure COMLR, Num. Math. 12, 369-376(1968) by Martin
0353      C-----and Wilkinson.
0354      C-----Handbook for Auto. Comp., Vol. III-Lineer Algebra, 396-403(1971).
0355      C-----The unitary analogue substitutes the QR algorithm of Francis
0356      C----- (Comp. Jour. 4, 332-345(1962)) for the LR Algorithm.
0357      C-----This subroutine finds the eigenvalues of a complex
0358      C-----upper Hessenberg matrix by the QR method.
0359      C-----On input:
0360      C-----NM must be set to the row dimension of two-dimensional
0361      C-----array parameters as declared in the calling program
0362      C-----dimension statement;
0363      C-----N is the order of the matrix;
0364      C-----LOW and IGH are integers determined by the balancing
0365      C-----subroutine CBAL. If CBAL has not been used,
0366      C-----set LOW=1, IGH=N;
0367      C-----HR and HI contain the real and imaginary parts,
0368      C-----respectively, of the complex upper Hessenberg matrix.
0369      C-----their lower triangles below the subdiagonal contain
0370      C-----information about the unitary transformations used in
0371      C-----the reduction by QORTH, if performed.
0372      C-----On output:
0373      C-----The upper Hessenberg portions of HR and HI have been
0374      C-----destroyed. Therefore, they must be saved before
0375      C-----calling COMQR if subsequent calculation of
0376      C-----eigenvectors is to be performed;
0377      C-----0378      C-----0379      C-----0380      C-----0381      C-----0382      C-----0383      C-----0384      C-----0385      C

```

```

C WR and WI contain the real and imaginary parts,
C respectively, of the eigenvalues. If an error
C exits is made, the eigenvalues should be correct
C for indices IERR+1,...,N

0386 C
0387 C
0388 C
0389 C
0390 C
0391 C IERR is set to
0392 C      ZERO for normal return,
0393 C      if the J-th eigenvalue has not been
0394 C      determined after 30 iterations.

0395 C
0396 C arithmetic is real except for the replacement of the ALGOL
0397 C procedure CDIV by complex division and use of the subroutines
0398 C CDSQRT and DCPLX in computing complex square roots.

0399 C
0400 C Questions and comments should be directed to B. S. Garbow,
0401 C Applied Mathematics Division, Argonne National Laboratory
0402 C
0403 C
0404 C
0405 C      MACHEP is a machine dependent parameter specifying
0406 C      the relative precision of floating point arithmetic.
0407 C      MACHEP = 16.0D0**(-143) for long form arithmetic
0408 C      on S360
0409 C      DATA MACHEP/1.421D-14/
0410 C
0411 C      IERR = 0
0412 C      IF (LOW .EQ. IGH) GO TO 180
0413 C      ::::: create real subdiagonal elements :::::::
0414 C      L = LOW + 1
0415 C
0416 DO 170 I = L, IGH
0417   LL = MIN(I,I+1,IGH)
0418   01 IF (HI(I,I-1) .EQ. 0.0D0) GO TO 170
0419   01 NORM = CDABS(DCMPLX(HR(I,1-1),HI(I,1-1)))
0420   01 YR = HR(I,I-1) / NORM
0421   01 YI = HI(I,I-1) / NORM
0422   01 HR(I,I-1) = NORM
0423   01 HI(I,I-1) = 0.0D0
0424   01
0425   01 DO 155 J = I, IGH
0426   02   SI = YR * HI(I,J) - YI * HR(I,J)
0427   02   HR(I,J) = YR * HR(I,J) + YI * HI(I,J)
0428   02   HI(I,J) = SI
0429   02
0430   02 155 CONTINUE
0431   01
0432   02 DO 160 J = LOW, LL
0433   02   SI = YR * HI(J,1) + YI * HR(J,1)
0434   02   HR(J,1) = YR * HR(J,1) - YI * HI(J,1)
0435   02   HI(J,1) = SI
0436   02 160 CONTINUE
0437   01 170 CONTINUE
0438   01
0439   01 180 DO 200 I = 1, N
0440   01   IF (I .GE. LOW .AND. I .LE. IGH) GO TO 200
0441   01   WR(I) = HR(I,I)
0442   01   WI(I) = HI(I,I)

```

```

0443.01      200 CONTINUE
0444.01      C   EN = 1GH
0445          EN = 1GH
0446          TR = 0.0DO
0447          TI = 0.0DO
C   ::::::: Search for next eigenvalue :::::::
0448          220 IF (EN .LT. LOW) GO TO 1001
0449          ITS = 0
0450
0451          ENM1 = EN - 1
C   ::::::: Look for single small sub-diagonal element
0452          FOR L=EN STEP -1 UNTIL LOW DO :::::::
0453          240 DO 260 LL = LOW, EN
0454          L = EN + LOW - LL
0455.01
0456.01
0457.01      X   IF (DABS(HR(L,L-1)) .LE.
0458.01      X   MACHEP # (DABS(HR(L-1,L-1)) + DABS(HI(L-1,L-1))
0459.01      X   + DABS(HR(L,L)) + DABS(HI(L,L))) GO TO 300
0460.01      260 CONTINUE
0461.01      C   ::::::: Form shift :::::::
0462          300 IF (L .EQ. EN) GO TO 460
0463          IF (ITS .EQ. 30) GO TO 1000
0464          IF (ITS .EQ. 10 .OR. ITS .EQ. 20) GO TO 320
0465          SR = HR(EN,EN)
0466          SI = HI(EN,EN)
0467          XR = HR(ENM1,EN) * HR(EN,ENM1)
0468          XI = HI(ENM1,EN) * HR(EN,ENM1)
0469          IF (XR .EQ. 0.0DO .AND. XI .EQ. 0.0DO) GO TO 340
0470          YR = (HR(ENM1,ENM1) - SR) / 2.0DO
0471          YI = (HI(ENM1,ENM1) - SI) / 2.0DO
0472          Z3 = DCMLX(YR**2-YI**2+XR, 2.0D0*YR*YI+XI)
0473          ZZR = DREAL(Z3)
0474          Z2I = DIMAG(Z3)
0475          IF (YR * ZZR + YI * Z2I .GE. 0.0DO) GO TO 310
0476          ZZR = -ZZR
0477          Z2I = -Z2I
0478          310 Z3 = DCMLX(XR, XI) / DCMLX(YR+ZZR, YI+Z2I)
0479          SR = SR - DREAL(Z3)
0480          SI = SI - DIMAG(Z3)
0481          GO TO 340
0482          C   ::::::: Form exceptional shift :::::::
0483          320 SR = DABS(HR(EN,ENM1)) + DABS(HR(ENM1,EN-2))
0484          SI = 0.0DO
0485          C   ::::::: Form shift :::::::
0486          340 DO 360 I = LOW, EN
0487.01          HR(I,I) = HR(I,I) - SR
0488.01          HI(I,I) = HI(I,I) - SI
0489.01          360 CONTINUE
0490.01      C   ::::::: Reduce to triangle (rows) :::::::
0491          TR = TR + SR
0492          TI = TI + SI
0493          ITS = ITS + 1
0494          C   ::::::: Reduce to triangle (rows) :::::::
0495          LP1 = L + 1
0496          C   DO 500 I = LP1, EN
0497          SR = HR(I,I-1)
0498.01          HR(I,I-1) = 0.0DO
0499.01

```

```

0500.01      NORM = DSGRT(HR(I-1,I-1)*HR(I-1,I-1)+HI(I-1,I-1)*HI(I-1,I-1)) AT 828
0501.01      X     XR = HR(I-1,I-1) / NORM
0502.01      WR(I-1) = XR
0503.01      XI = HI(I-1,I-1) / NORM
0504.01      XJ = XI
0505.01      WI(I-1) = XI
0506.01      HR(I-1,I-1) = NORM
0507.01      HI(I-1,I-1) = O.ODO
0508.01      HI(I-1,I-1) = SR / NORM
0509.01      C     DO 490 J = 1, EN
0510.01      YR = HR(I-1,J)
0511.02      YI = HI(I-1,J)
0512.02      ZZR = HR(I,J)
0513.02      Z2I = HI(I,J)
0514.02      HR(I-1,J) = XR * YR + XI * YI + HI(I,J-1) * ZZR
0515.02      HI(I-1,J) = XR * YI - XI * YR + HI(I,J-1) * Z2I
0516.02      HR(I,J) = XR * ZZR - XI * Z2I - HI(I,J-1) * YR
0517.02      HI(I,J) = XR * Z2I + XI * ZZR - HI(I,J-1) * YI
0518.02      490  CONTINUE
0519.02      490  CONTINUE
0520.02      C     500 CONTINUE
0521.01      C
0522.01      C     SJ = HI(EN,EN)
0523      IF (SJ .EQ. 0.ODO) GO TO 540
0524      NORM = CDABSI(CHPLX(HR(EN,EN), SI))
0525      SR = HR(EN,EN) / NORM
0526      SI = SI / NORM
0527      HR(EN,EN) = O.ODO
0528      HI(EN,EN) = NORM
0529      540  DD 600 J = LP1, EN
0530      C     ::::::: Inverse operation (columns) :::::::
0531      540  DD 600 J = LP1, EN
0532.01      XJ = WR(J-1)
0533.01      XI = WI(J-1)
0534.01      C     DO 580 I = L, J
0535.01      YR = HR(I,J-1)
0536.02      YI = O.ODO
0537.02      ZZR = HR(I,J)
0538.02      Z2I = HI(I,J)
0539.02      IF (I .EQ. J) GO TO 560
0540.02
0541.02
0542.02      VI = HI(I,J-1)
      HI(I,J-1) = XR * YI + XI * YR + HI(J,J-1) * Z2I
      HR(I,J-1) = XR * YR - XI * YI + HI(J,J-1) * ZZR
0543.02      HI(I,J) = XR * ZZR + XI * Z2I - HI(J,J-1) * YR
0544.02      HI(I,J) = XI * Z2I - XI * ZZR - HI(J,J-1) * YI
0545.02      580  CONTINUE
0546.02
0547.02      C     600 CONTINUE
0548.01      C     IF (SI .EQ. 0.ODO) GO TO 240
0549.01      C
0550      C     DO 630 I = L, EN
0551      YR = HR(I,EN)
0552.01      YI = HI(I,EN)
0554.01      HR(I,EN) = SR * YR - SI * YI
0555.01      HI(I,EN) = SR * YI + SI * YR
0556.01

```

```

0557.01    630 CONTINUE
0558.01    C   GO TO 240
0559.01    C   ..... A root found .....,.
0560.01    C   660 WR(EN) = HR(EN,EN) + TR
0561.01    C   WI(EN) = HI(EN,EN) + TI
0562.01    C   EN = ENM1
0563.01    C   GO TO 220
0564.01    C   ..... Set error -- no convergence to an
0565.01    C   eigenvalue after 30 iterations :::::::
0566.01    C   1000 JERR = EN
0567.01    1000 RETURN
0568.01    C   ..... last card of COMAR :::::::
0569.01    C   END
0570.01

0571 C***** SUBROUTINE COMGR2(NM,N,LOW,IGH,DRTR,ORTI,HR,HI,WR,WI,ZR,ZI,IERR)
0572 C----- INTEGER I,J,K,L,M,N,EN,II,JJ,LL,NM,NN,IGH,IP1,
0573 C----- X,ITS,LOW,LP1,ENM1,IEND,IERR
0574 C----- REAL*B HR(NM,N),HI(NM,N),WR(N),WI(N),ZR(NM,NN),ZI(NM,N),
0575 C----- X,DRTR(IGH),ORTI(IGH)
0576 C----- REAL*B SI,SR,TR,XI,XR,YI,YR,ZZI,ZZR,NORM,MACHEP
0577 C----- REAL*B DSQRT,CDABS,DABS
0578 C----- INTEGER MINO
0579 C----- COMPLEX*16 Z3
0580 C----- COMPLEX*16 CDSGRT,DCMPLX
0581 C----- REAL*B DREAL,DIMAG
0582 C----- ..... Statement functions enable extraction of real and
0583 C----- imaginary parts of double precision complex numbers :::::::
0584 C----- Statement functions enable extraction of real and
0585 C----- imaginary parts of double precision complex numbers :::::::
0586 C----- DREAL(Z3) = Z3
0587 C----- DIMAG(Z3) = (0.0D0,-1.0D0) * Z3
0588 C
0589 C This subroutine is a translation of a unitary analogue of the
0590 C ALGOL procedure COMLR2, Num. Math. 16, 181-204(1970) by Peters
0591 C and Wilkinson.
0592 C Handbook for Auto. Comp., Vol. II-Linear Algebra, 372-395(1971).
0593 C The unitary analogue substitutes the QR algorithm of Francis
0594 C (Comp. Jour. 4, 332-345(1962)) for the LR algorithm.
0595 C
0596 C This subroutine finds the eigenvalues and eigenvectors
0597 C of a complex upper Hessenberg matrix by the QR
0598 C method. The eigenvectors of a complex general matrix
0599 C can also be found if CORTH has been used to reduce
0600 C this general matrix to Hessenberg form.
0601 C
0602 C On input:
0603 C   NM must be set to the row dimension of two-dimensional
0604 C   array parameters as declared in the calling program
0605 C   DIMENSION statement;
0606 C
0607 C   N is the order of the matrix;
0608 C
0609 C   LOW and IGH are integers determined by the balancing
0610 C

```

```

0611 C subroutine CBAL, 1f CBAL has not been used,
0612 C set LOW=1, IGH=N;
0613 C
0614 C ORTR and ORTI contain information about the unitary trans-
0615 C formations used in the reduction by CORTH, if performed;
0616 C only elements LOW through IGH are used. If the eigenvectors
0617 C of the Hessenberg matrix are desired, set ORTR(J) and
0618 C ORTI(J) to 0. ODO for these elements;
0619 C
0620 C HR and HI contain the real and imaginary parts,
0621 C respectively, of the complex upper Hessenberg matrix.
0622 C Their lower triangles below the subdiagonals contain further
0623 C information about the transformations which were used in the
0624 C reduction by CORTH, if performed. If the eigenvectors of
0625 C the Hessenberg matrix are desired, these elements may be
0626 C arbitrary.
0627 C
0628 C On output:
0629 C
0630 C ORTR, ORTI, and the upper Hessenberg portions of HR and HI
0631 C have been destroyed.
0632 C
0633 C HR and WI contain the real and imaginary parts,
0634 C respectively, of the eigenvalues. If an error
0635 C exit is made, the eigenvalues should be correct
0636 C for indices IERR+1,...,N;
0637 C
0638 C ZR and ZI contain the real and imaginary parts,
0639 C respectively, of the eigenvector. The eigenvectors
0640 C are unnormalized. If an error exit is made, none of
0641 C the eigenvectors has been found;
0642 C
0643 C IERR is set to
0644 C zero for normal return,
0645 C if the J-th eigenvalue has not been
0646 C determined after 30 iterations.
0647 C
0648 C Arithmetic is real except for the replacement of the ALGOL_
0649 C procedure CDIV by complex division and use of the subroutines
0650 C CDSQRT and DCMPLEX in computing complex square roots.
0651 C
0652 C Questions and comments should be directed to B. S. Garbow,
0653 C Applied Mathematics Division, Argonne National Laboratory
0654 C
0655 C
0656 C
0657 C
0658 C
0659 C
0660 C
0661 C
0662 C
0663 C
0664 C
0665 C
0666 C
0667 C
0668 C
0669 C
0670 C
0671 C
0672 C
0673 C
0674 C
0675 C
0676 C
0677 C
0678 C
0679 C
0680 C
0681 C
0682 C
0683 C
0684 C
0685 C
0686 C
0687 C
0688 C
0689 C
0690 C
0691 C
0692 C
0693 C
0694 C
0695 C
0696 C
0697 C
0698 C
0699 C
0700 C
0701 C
0702 C
0703 C
0704 C
0705 C
0706 C
0707 C
0708 C
0709 C
0710 C
0711 C
0712 C
0713 C
0714 C
0715 C
0716 C
0717 C
0718 C
0719 C
0720 C
0721 C
0722 C
0723 C
0724 C
0725 C
0726 C
0727 C
0728 C
0729 C
0730 C
0731 C
0732 C
0733 C
0734 C
0735 C
0736 C
0737 C
0738 C
0739 C
0740 C
0741 C
0742 C
0743 C
0744 C
0745 C
0746 C
0747 C
0748 C
0749 C
0750 C
0751 C
0752 C
0753 C
0754 C
0755 C
0756 C
0757 C
0758 C
0759 C
0760 C
0761 C
0762 C
0763 C
0764 C
0765 C
0766 C
0767 C
0768 C
0769 C
0770 C
0771 C
0772 C
0773 C
0774 C
0775 C
0776 C
0777 C
0778 C
0779 C
0780 C
0781 C
0782 C
0783 C
0784 C
0785 C
0786 C
0787 C
0788 C
0789 C
0790 C
0791 C
0792 C
0793 C
0794 C
0795 C
0796 C
0797 C
0798 C
0799 C
0800 C
0801 C
0802 C
0803 C
0804 C
0805 C
0806 C
0807 C
0808 C
0809 C
0810 C
0811 C
0812 C
0813 C
0814 C
0815 C
0816 C
0817 C
0818 C
0819 C
0820 C
0821 C
0822 C
0823 C
0824 C
0825 C
0826 C
0827 C
0828 C
0829 C
0830 C
0831 C
0832 C
0833 C
0834 C
0835 C
0836 C
0837 C
0838 C
0839 C
0840 C
0841 C
0842 C
0843 C
0844 C
0845 C
0846 C
0847 C
0848 C
0849 C
0850 C
0851 C
0852 C
0853 C
0854 C
0855 C
0856 C
0857 C
0858 C
0859 C
0860 C
0861 C
0862 C
0863 C
0864 C
0865 C
0866 C
0867 C
0868 C
0869 C
0870 C
0871 C
0872 C
0873 C
0874 C
0875 C
0876 C
0877 C
0878 C
0879 C
0880 C
0881 C
0882 C
0883 C
0884 C
0885 C
0886 C
0887 C
0888 C
0889 C
0890 C
0891 C
0892 C
0893 C
0894 C
0895 C
0896 C
0897 C
0898 C
0899 C
0900 C
0901 C
0902 C
0903 C
0904 C
0905 C
0906 C
0907 C
0908 C
0909 C
0910 C
0911 C
0912 C
0913 C
0914 C
0915 C
0916 C
0917 C
0918 C
0919 C
0920 C
0921 C
0922 C
0923 C
0924 C
0925 C
0926 C
0927 C
0928 C
0929 C
0930 C
0931 C
0932 C
0933 C
0934 C
0935 C
0936 C
0937 C
0938 C
0939 C
0940 C
0941 C
0942 C
0943 C
0944 C
0945 C
0946 C
0947 C
0948 C
0949 C
0950 C
0951 C
0952 C
0953 C
0954 C
0955 C
0956 C
0957 C
0958 C
0959 C
0960 C
0961 C
0962 C
0963 C
0964 C
0965 C
0966 C
0967 C
0968 C
0969 C
0970 C
0971 C
0972 C
0973 C
0974 C
0975 C
0976 C
0977 C
0978 C
0979 C
0980 C
0981 C
0982 C
0983 C
0984 C
0985 C
0986 C
0987 C
0988 C
0989 C
0990 C
0991 C
0992 C
0993 C
0994 C
0995 C
0996 C
0997 C
0998 C
0999 C
0999 C

```

```

0668.02      ZR(1,J) = 0. ODO          AT 47
0669.02      ZI(1,J) = 0. ODO          AT 58
0670.02      IF (J .EQ. J) ZR(I,J) = 1. ODO  AT 67
0671.02
0672.02 C   100 CONTINUE   Form the matrix of accumulated transformations
0673.02 C   From the information left by CORTH ::::::::::::;
0674      IEND = IGH - LDW - 1          AT 84
0675      IF (IEND) 1B0, 1C5, 1C5
0676      C   ::::::::::::: FDR I=IGH-1 STEP -1 UNTIL LDW+1 DO --- ::::::::::::;
0677      105 DO 140 IJ = 1, IEND          AT 90
0678.01      1 = IGH - II
0679.01      IF (DTR(1) .EQ. 0. ODO .AND. ORT1(1) .EQ. 0. ODO) GO TO 140  AT 94
0680.01      IF (HR(1,1-1) .EQ. 0. ODO .AND. HI(1,1-1) .EQ. 0. ODO) GO TO 140  AT 103
0681.01 C   ::::::::::::: Norm below is negative of H formed in CORTH ::::::::::::;
0682.01      NORM = HR(1,1-1) * ORTR(1) + HI(1,1-1) * ORT1(1)          AT 107
0683.01      IP1 = 1 + I
0684.01 C
0685.01      DO 110 K = IP1, IGH          AT 123
0686.02      ORTR(K) = HR(K,1-1)          AT 123
0687.02      ORT1(K) = HI(K,1-1)
0688.02      CONTINUE
0689.02 C   110
0690.01      DO 130 J = 1, IGH          AT 140
0691.02      SR = 0. ODO
0692.02      SI = 0. ODO
0693.02 C
0694.02      DO 115 K = 1, IGH          AT 140
0695.03      SR = SR + ORTR(K) * ZR(K,J) + ORT1(K) * ZI(K,J)          AT 140
0696.03      SI = SI + ORTR(K) * ZI(K,J) - ORT1(K) * ZR(K,J)          AT 140
0697.03      115 CONTINUE
0698.03 C
0699.02      SR = SR / NORM          AT 140
0700.02      SI = SI / NORM          AT 140
0701.02 C
0702.02      DO 120 K = 1, IGH          AT 140
0703.03      ZR(K,J) = ZR(K,J) + SR * ORTR(K) - SI * ORT1(K)          AT 140
0704.03      ZI(K,J) = ZI(K,J) + SR * ORT1(K) + SI * ORTR(K)          AT 140
0705.03      120 CONTINUE
0706.03 C
0707.02      130 CONTINUE
0708.02 C
0709.01      140 CONTINUE
0710.01 C   ::::::::::::: Create real subdiagonal elements ::::::::::::;
0711      150 L = LDW + 1
0712 C
0713      DO 170 J = L, IGH          AT 140
0714.01      LL = MIN(J+1, IGH)          AT 140
0715.01      IF (HI(1,1-1) .EQ. 0. ODO) GO TO 170  AT 140
0716.01      NORM = CDABS(DCMPLX(HR(1,1-1),HI(1,1-1)))  AT 140
0717.01      VR = HR(1,1-1) / NORM  AT 140
0718.01      YI = HI(1,1-1) / NORM  AT 140
0719.01      HR(1,1-1) = NORM  AT 140
0720.01      HI(1,1-1) = 0. ODO  AT 140
0721.01 C
0722.01      DO 155 J = 1, N          AT 140
0723.02      SI = YR * HI(1,J) - YI * HR(1,J)          AT 140
0724.02      HR(1,J) = YR * HR(1,J) + YI * HI(1,J)          AT 140

```

20:40:07

SUN, AUG 06 1989

```

0725.02      HI(I,J) = SI          AT 552
0726.02      155      CONTINUE    AT 563
0727.02      C          DD 160 J = 1, LL
0728.01      SI = YR * HI(J,1) + YI * HR(J,1)
0729.02      HR(J,1) = YR * HR(J,1) - YI * HI(J,1)
0730.02      HI(J,1) = SI
0731.02      160      CONTINUE
0732.02      C          DD 165 J = LDW, IGH
0733.02      SI = YR * ZI(J,1) + YI * ZR(J,1)
0734.01      ZR(J,1) = YR * ZR(J,1) - YI * ZI(J,1)
0735.02      165      CONTINUE
0736.02      C          DD 160 J = 1, LL
0737.02      165      CONTINUE
0738.02      C          DD 165 J = LDW, IGH
0739.02      170      CONTINUE
0740.01      C          ::::::: Store roots isolated by CBAL :::::::
0741.01      ::::::: 180 DO 200 I = 1, N
0742.02      ::::::: IF (I .GE. LDW .AND. I .LE. IGH) GO TO 200
0743.01      ::::::: WR(I) = HR(I,I)
0744.01      ::::::: WR(I) = HI(I,I)
0745.01      ::::::: W1(I) = HI(I,I)
0746.01      200      CONTINUE
0747.01      C          EN = IGH
0748.01      ::::::: TR = 0.0D0
0749.01      ::::::: TI = 0.0D0
0750.01      C          ::::::: Search for next eigenvalue :::::::
0751.01      ::::::: 220 IF (EN .LT. LDW) GO TO 680
0752.01      ::::::: ITS = 0
0753.01      ::::::: ENM1 = EN - I
0754.01      ::::::: Look for single small sub-diagonal element
0755.01      ::::::: for L=EN STEP -1 UNTIL LDW DO -- :::::::
0756.01      ::::::: 240 DO 260 LL = LDW, EN
0757.01      ::::::: L = EN + LDW - LL
0758.01      ::::::: IF (L .EQ. LDW) GO TO 300
0759.01      ::::::: 1F (DABSR(HR(L,L-1)), LE,
0760.01      ::::::: X     MACHEP * (DABS(HR(L-1,L-1)) + DABS(HI(L-1,L-1)))
0761.01      ::::::: X     + DABS(HR(L,L)) + DABS(HI(L,L))) GO TO 300
0762.01      ::::::: 260      CONTINUE
0763.01      ::::::: 300      IF (L .EQ. EN) GO TO 640
0764.01      ::::::: 300      IF (ITS .EQ. 30) GO TO 1000
0765.01      ::::::: 1F (ITS .EQ. 10 .OR. ITS .EQ. 20) GO TO 320
0766.01      ::::::: SR = HR(EN,EN)
0767.01      ::::::: SI = HI(EN,EN)
0768.01      ::::::: XR = HR(ENM1,EN) * HR(EN,ENM1)
0769.01      ::::::: XI = HI(ENM1,EN) * HR(EN,ENM1)
0770.01      ::::::: 1F (XR .EQ. 0.0D0 .AND. XI .EQ. 0.0D0) GO TO 340
0771.01      ::::::: YR = (HR(ENM1,ENM1) - SR) / 2.0D0
0772.01      ::::::: YI = (HI(ENM1,ENM1) - SI) / 2.0D0
0773.01      ::::::: Z3 = CDGORT(DCMPLX(YR**2-YI**2+XR, 2, 0D0*YR*YI+XI))
0774.01      ::::::: Z2R = DREAL(Z3)
0775.01      ::::::: Z2I = DIMAG(Z3)
0776.01      ::::::: 1F (YR * ZZR + YI * ZZI .GE. 0.0D0) GO TO 310
0777.01      ::::::: ZZR = -ZZR
0778.01      ::::::: ZZI = -ZZI
0779.01      ::::::: 310 Z3 = DCMPLX(XR, XI) / DCMPLX(YR+ZZR, YI+ZZI)

```

```

SR = SR - DREAL(Z3)
SI = SI - DIMAG(Z3)
GO TO 340
C      ::::::: Form exceptional shift :::::::
0785      320 SR = DABS(HR(EN,ENM1)) + DABS(HR(ENM1,EN-2))
SI = O.ODO
0788      C      340 DO 360 J = LOW, EN
0789      01      HR(I,J) = HR(I,I) - SR
0791      01      HI(I,J) = HI(I,I) - SI
0792      01      360 CONTINUE
0793      01      C      TR = TR + SR
0794      01      TI = TI + SI
0795      01      ITS = ITS + 1
0796      C      ::::::: Reduce to triangle (rows) :::::::
0797      C      LP1 = L + 1
0798      C      ::::::: ::::::: ::::::: ::::::: ::::::: :::::::
0799      C      ::::::: ::::::: ::::::: ::::::: ::::::: :::::::
0800      DO 500 I = LP1, EN
0801      01      SR = HR(I,I-1)
0802      01      HR(I,I-1) = O.ODO
0803      01      NDIM = DSQRT(HR(I-1,I-1)*HR(I-1,I-1)+HI(I-1,I-1)*HI(I-1,I-1))
0804      01      X      XR = HR(I-1,I-1) / NORM
0805      01      WR(I-1) = XR
0806      01      XI = HI(I-1,I-1) / NORM
0807      01      WI(I-1) = XI
0808      01      HR(I-1,I-1) = NORM
0809      01      HI(I-1,I-1) = O.ODO
0810      01      HII(I,I-1) = SR / NORM
0811      01      C      DO 490 J = 1, N
0812      01      01      DO 490 J = 1, N
0813      01      02      VR = HR(I-1,J)
0814      02      02      YI = HI(I-1,J)
0815      02      02      ZZR = HR(I,J)
0816      02      02      ZII = HI(I,J)
0817      02      02      HR(I-1,J) = XR * VR + XI * YI + HI(I,I-1) * ZZR
0818      02      02      HI(I-1,J) = XR * YI - XI * VR + HI(I,I-1) * ZII
0819      02      02      HR(I,J) = XR * ZZR - XI * ZII - HI(I,I-1) * YR
0820      02      02      HII(I,J) = XR * ZII + XI * ZZR - HI(I,I-1) * YI
0821      02      02      490  CONTINUE
0822      02      C      500 CONTINUE
0823      02      C      500 CONTINUE
0824      01      C      500 CONTINUE
0825      01      C      500 CONTINUE
0826      C      SI = HI(EN,EN)
0827      C      IF (SI .EQ. O.ODO) GO TO 540
0828      C      NORM = CDARS(DCMPLX(HR(EN,EN),SI))
0829      C      SR = HR(EN,EN) / NORM
0830      C      SI = SI / NORM
0831      C      HR(EN,EN) = NORM
0832      C      HI(EN,EN) = O.ODO
0833      C      IF (EN .EQ. N) GO TO 540
0834      C      IP1 = EN + 1
0835      C      DO 520 J = IP1, N
0836      C      VR = HR(EN,J)
0837      01      YI = HI(EN,J)
0838      01      ::::::: ::::::: ::::::: ::::::: ::::::: :::::::
AT 1092
AT 1102
AT 1125
AT 1126
AT 1155
AT 1159
AT 1171
AT 1184
AT 1197
AT 1198
AT 1204
AT 1210
AT 1212
AT 1215
AT 1226
AT 1237
AT 1249
AT 1251
AT 1263
AT 1275
AT 1283
AT 1298
AT 1304
AT 1319
AT 1325
AT 1338
AT 1351
AT 1354
AT 1365
AT 1377
AT 1389
AT 1401
AT 1412
AT 1423
AT 1461
AT 1499
AT 1536
AT 1573
AT 1574
AT 1575
AT 1586
AT 1590
AT 1609
AT 1622
AT 1628
AT 1639
AT 1650
AT 1655
AT 1658
AT 1670
AT 1681

```

```

0839.01          HR(EN,J) = SR * YR + SI * YI
0840.01          HI(EN,J) = SR * YI - SI * YR
0841.01          520 CONTINUE
0842.01          C : Inverse opération (columns)
0843.01          540 DO 600 J = LP1, EN
0844.01          XR = WR(J-1)
0845.01          XI = WI(J-1)
0846.01          C
0847.01          DO 580 I = 1, J
0848.02          YR = HR(I,J-1)
0849.02          YI = 0.0D0
0850.02          ZZR = HR(I,J)
0851.02          ZZI = HI(I,J)
0852.02          IF (I .EQ. J) GO TO 580
0853.02          YI = HI(I,J-1)
0854.02          HI(I,J-1) = XR * YI + XI * YR + HI(J,J-1) * ZZR
0855.02          HR(I,J-1) = XR * YR - XI * YI + HI(J,J-1) * ZZR
0856.02          HR(I,J) = XR * ZZR + XI * ZZI - HI(J,J-1) * YR
0857.02          HI(I,J) = XR * ZZI - XI * ZZR - HI(J,J-1) * YI
0858.02          CONTINUE
0859.02          C
0860.01          DO 590 I = L0W, IGH
0861.02          YR = 2R(I,J-1)
0862.02          YI = 2I(I,J-1)
0863.02          ZZR = 2R(I,J)
0864.02          ZZI = 2I(I,J)
0865.02          ZR(I,J-1) = XR * YR - XI * YI + HI(J,J-1) * ZZR
0866.02          ZI(I,J-1) = XR * YI + XI * YR + HI(J,J-1) * ZZR
0867.02          ZR(I,J) = XR * ZZR + XI * ZZI - HI(J,J-1) * YR
0868.02          ZI(I,J) = XR * ZZI - XI * ZZR - HI(J,J-1) * YI
0869.02          CONTINUE
0870.02          C
0871.01          600 CONTINUE
0872.01          C
0873.01          IF (SI .EQ. 0.0D0) GO TO 240
0874.01          C
0875.01          DO 630 I = 1, EN
0876.01          YR = HR(I,EN)
0877.01          YI = HI(I,EN)
0878.01          HR(I,EN) = SR * YR - SI * YI
0879.01          HI(I,EN) = SR * YI + SI * YR
0880.01          630 CONTINUE
0881.01          C
0882.01          DO 640 I = L0W, IGH
0883.01          YR = 2R(I,EN)
0884.01          YI = 2I(I,EN)
0885.01          ZR(I,EN) = SR * YR - SI * YI
0886.01          ZI(I,EN) = SR * YI + SI * YR
0887.01          640 CONTINUE
0888.01          C
0889.01          DO 60 TO 240
0890.01          C : A root found
0891.01          660 HR(EN,EN) = HR(EN,EN) + TR
0892.01          WR(EN) = HR(EN,EN)
0893.01          HI(EN,EN) = HI(EN,EN) + TI
0894.01          WI(EN) = HI(EN,EN)
0895.01          EN = ENM1

```

```

0856      GO TO 220
0857      C      ::::::: All roots found. Backsubstitute to find
0858      C      vectors of upper triangular form :::::::
0859      C      680 NORM = 0. ODO
0900      C      DO 720 I = 1, N
0901      C      DO 720 J = 1, N
0902      C      DO 720 NORM = NORM + DABS(HI(I,J)) + DABS(HI(J,I))
0903.01    0904.02    0905.02    0906.02    0907    IF (N .EQ. 1 .OR. NORM .EQ. 0. ODO) GO TO 1001
0908      C      ::::::: FOR EN=N STEP -1 UNTIL 2 DO -- :::::::
0909      C      DO 800 NN = 2, N
0910.01    0911.01    0912.01    0913.01    0914.01    0915.01    0916.02    0917.02    0918.02    0919.02    0920.02    0921.02    0922.02    0923.03    0924.03    0925.03    0926.03    0927.02    0928.02    0929.02    0930.02    0931.02    0932.02    0933.02    0934.02    0935.01    0936.01    0937    0938    0939    0940.01    0941.01    0942.01    0943.01    0944.02    0945.02    0946.02    0947.02    0948.01    0949.01    0950.01    0951.01    0952
          DO 720 I = 1, N
          NORM = NORM + DABS(HR(I,J)) + DABS(HR(J,I))
          720 CONTINUE
          IF (N .EQ. 1 .OR. NORM .EQ. 0. ODO) GO TO 1001
          DO 800 NN = 2, N
          EN = N + 2 - NN
          XR = WR(EN)
          XI = WI(EN)
          ENM1 = EN - 1
          ::::::: FOR I=EN-1 STEP -1 UNTIL 1 DO -- :::::::
          DO 780 II = 1, ENM1
          I = EN - II
          ZZR = HR(I,EN)
          ZZI = HI(I,EN)
          IF (I .EQ. ENM1) GO TO 760
          IP1 = I + 1
          DO 740 J = IP1, ENM1
          ZZR = ZZR + HR(I,J) * HI(J,EN) - HI(J,J) * HI(J,EN)
          ZZI = ZZI + HR(I,J) * HI(J,EN) + HI(I,J) * HR(J,EN)
          CONTINUE
          YR = XR - WR(1)
          YI = XI - WI(1)
          IF (YR .EQ. 0. ODO .AND. YI .EQ. 0. ODO) YR = MACHEP * NORM
          Z3 = DCMPXL(ZZR,ZZI) / DCMPXL(YR,YI)
          HR(1,EN) = DREAL(Z3)
          HI(1,EN) = DIMAG(Z3)
          CONTINUE
          YR = XR - WR(1)
          YI = XI - WI(1)
          IF (YR .EQ. 0. ODO .AND. YI .EQ. 0. ODO) YR = MACHEP * NORM
          Z3 = DCMPXL(ZZR,ZZI) / DCMPXL(YR,YI)
          HR(1,EN) = DREAL(Z3)
          HI(1,EN) = DIMAG(Z3)
          CONTINUE
          800 CONTINUE
          ::::::: End backsubstitution :::::::
          ENM1 = N - 1
          ::::::: Vectors of isolated roots :::::::
          DO 840 I = 1, ENM1
          IF (I .GE. LOW .AND. I .LE. HIGH) GO TO 840
          IP1 = I + 1
          DO 820 J = IP1, N
          ZR(I,J) = HR(I,J)
          ZI(I,J) = HI(I,J)
          CONTINUE
          840 CONTINUE
          ::::::: Multiply by transformation matrix to give
          ::::::: vectors of original full matrix.
          FOR J=N STEP -1 UNTIL LOW+1 DO -- :::::::
          DO 880 JJ = LOW, ENM1
          AT 2389
          AT 2390
          AT 2394
          AT 2404
          AT 2416
          AT 2439
          AT 2441
          AT 2452
          AT 2462
          AT 2467
          AT 2474
          AT 2482
          AT 2485
          AT 2493
          AT 2496
          AT 2507
          AT 2518
          AT 2522
          AT 2525
          AT 2536
          AT 2566
          AT 2596
          AT 2597
          AT 2607
          AT 2617
          AT 2635
          AT 2664
          AT 2675
          AT 2699
          AT 2700
          AT 2701
          AT 2705
          AT 2713
          AT 2721
          AT 2723
          AT 2735
          AT 2746
          AT 2757
          AT 2758
          AT 2759
        
```

```

0953.01      J = N + LOW - JJ          AT 2771
0954.01      NM = MIND(J-1, IGH)       AT 2777
0955.01      C
0956.01      DO 860 I = LOW, IGH      AT 2786
0957.02      ZZR = ZR(I,J)
0958.02      ZZI = ZI(I,J)
0959.02      C
0960.02      DO 860 K = LOW, NM        AT 2821
0961.03      ZZR = ZZR + ZR(I,K) * HR(K,J) - ZI(I,K) * HI(K,J)
0962.03      ZZI = ZZI + ZR(I,K) * HI(K,J) + ZI(I,K) * HR(K,J)
0963.03      CONTINUE                  AT 2833
0964.03      C
0965.02      ZR(I,J) = ZZR            AT 2863
0966.02      ZI(I,J) = ZZI            AT 2893
0967.02      E60 CONTINUE               AT 2894
0968.02      C
0969      GO TO 1001
0970      C : Set error -- no convergence to en
0971      C : eigenvalue after 30 iterations .....
0972      1000 TERR = EN                AT 2919
0973      1001 RETURN                  AT 2922
0974      C : last card of COMOR2 .....
0975      END

```

```

C***** SUBROUTINE CORTHNM,N,LOW,IGH,AR,AI,ORTR,ORT1
C
C-----
```

```

0976      INTEGER I,J,M,N,II,JJ,LA,MP,NM,IGH,KP1,LOW
0977      REAL*8 AR(NM,N),AI(NM,N),ORTR(IGH),ORT1(IGH)
0978      REAL*8 F,G,H,F1,FR,SCALE
0979      REAL*8 DSQRT,CDABS,DAB8
0980      COMPLEX*16 DCMPXL

```

0981 This subroutine is a translation of a complex analogue of  
the ALGOL procedure ORTHES, Num. Math. 12, 349-368(1968),  
by Martin and Wilkinson.

0982 Handbook for Auto. Comp., Vol. II-Linear Algebra, 339-358(1971).

0983 Given a complex general matrix, this subroutine  
reduces a submatrix situated in rows and columns  
LOW through IGH to upper Hessenberg form by  
unitary similarity transformations.

0984 On input:

0985 NM must be set to the row dimension of two-dimensional  
array parameters as declared in the calling program  
0986 DIMENSION statement;  
0987 C  
0988 C  
0989 C  
0990 C  
0991 C  
0992 C  
0993 C  
0994 C  
0995 C  
0996 C  
0997 C  
0998 C  
0999 C  
1000 C  
1001 C  
1002 C  
1003 C  
1004 C  
1005 C  
1006 C

0988 C  
0989 C  
1000 C  
1001 C  
1002 C  
1003 C  
1004 C  
1005 C  
1006 C

0988 C  
0989 C  
1000 C  
1001 C  
1002 C  
1003 C  
1004 C  
1005 C  
1006 C

```

1007      C AR and AI contain the real and imaginary parts,
1008      C respectively, of the complex input matrix.
1009      C
1010      C On output:
1011      C
1012      C AR and AI contain the real and imaginary parts,
1013      C respectively, of the Hessenberg matrix. Information
1014      C about the unitary transformations used in the reduction
1015      C is stored in the remaining triangles under the
1016      C Hessenberg matrix.
1017      C
1018      C ORTR and ORTI contain further information about the
1019      C transformations. Only elements LDW through IGH are used.
1020      C
1021      C Questions and comments should be directed to B. S. Gerbow,
1022      C Applied Mathematics Division, Argonne National Laboratory
1023      C
1024      C
1025      C
1026      C LA = IGH - 1
1027      C KP1 = LDW + 1
1028      C IF (LA .LT. KP1) GO TO 200
1029      C
1030      C DO 180 M = KP1, LA
1031      01      H = 0.0D0
1032      01      ORTR(M) = 0.0D0
1033      01      ORTI(M) = 0.0D0
1034      01      SCALE = 0.0D0
1035      01      C : : : : : Scale column (ALGOL TGL then not needed) : : : : :
1036      01      DO 90 I = M, IGH
1037      02      90      SCALE = SCALE + DABS(AI(I, M-1)) + DABS(AI(I, M-1))
1038      02      C
1039      01      C
1040      01      C IF (SCALE .EQ. 0.0D0) GO TO 180
1041      01      C MP = M + IGH
1042      01      C : : : : : FOR I=IGH STEP -1 UNTIL M DO --- : : : : :
1043      02      C DO 100 II = M, IGH
1044      02      C I = MP - II
1045      02      C ORTR(I) = AR(I, M-1) / SCALE
1046      02      C ORTI(I) = AI(I, M-1) / SCALE
1047      02      C H = H + ORTR(I) * ORTR(I) + ORTI(I) * ORTI(I)
1048      02      C
1049      01      C
1045      01      C G = DSORT(H)
1050      01      C F = CDABS(DCMPLX(ORTR(M), ORTI(M)))
1051      01      C IF (F .EQ. 0.0D0) GO TO 103
1052      01      C H = H + F * G
1053      01      C G = G / F
1054      01      C ORTR(M) = (1.0D0 + G) * ORTR(M)
1055      01      C ORTI(M) = (1.0D0 + G) * ORTI(M)
1056      01      C
1057      01      C
1058      01      C 103      ORTR(M) = G
1059      01      C AR(M, M-1) = SCALE
1060      01      C : : : : : Form (I-(U*UT)/H) * A : : : : :
1061      01      C 105      DO 130 J = M, N
1062      02      C FR = 0.0D0
1063      02      C FI = 0.0D0

```

```

1064.02 C ::::::::::: FOR I=1GH STEP -1 UNTIL M DO --- ::::::::::::
1065.02 DD 110 I1 = M, 1GH
1066.03 I = MP - 11
1067.03 FR = FR + ORTR(I) * AR(I,J) + ORTI(I) * AI(I,J)
1068.03 FI = FI + ORTR(I) * AI(I,J) - ORTI(I) * AR(I,J)
1069.03 110  CONTINUE
1070.03 C
1071.02 FR = FR / H
1072.02 F1 = F1 / H
1073.02 C
1074.02 DD 120 I = M, 1GH
1075.03 AR(I,J) = AR(I,J) - FR * ORTR(I) + FI * ORTI(I).
1076.03 AI(I,J) = AI(I,J) - FR * ORTI(I) - FI * ORTR(I)
1077.03 120  CONTINUE
1078.03 C
1079.02 130  CONTINUE
1080.02 C ::::::::::: Form (I-(UMUT)/H)*A*(I-(UMUT)/H) ::::::::::::
1081.01 DD 160 I = 1, 1GH
1082.02 FR = O.ODO
1083.02 FI = O.ODO
1084.02 C ::::::::::: FOR J=1GH STEP -1 UNTIL M DO --- ::::::::::::
1085.02 DD 140 JJ = M, 1GH
1086.03 J = MP - JJ
1087.03 FR = FR + ORTR(J) * AR(I,J) - ORTI(J) * AI(I,J)
1088.03 FI = FI + ORTR(J) * AI(I,J) + ORTI(J) * AR(I,J)
1089.03 140  CONTINUE
1090.03 C
1091.02 FR = FR / H
1092.02 F1 = F1 / H
1093.02 C
1094.02 DD 150 J = M, 1GH
1095.03 AR(I,J) = AR(I,J) - FR * ORTR(J) - FI * ORTI(J)
1096.03 AI(I,J) = AI(I,J) + FR * ORTI(J) - FI * ORTR(J)
1097.03 150  CONTINUE
1098.03 C
1099.02 160  CONTINUE
1100.02 C
1101.01 ORTR(M) = SCALE * ORTR(M)
1102.01 ORTI(M) = SCALE * ORTI(M)
1103.01 AR(M,M-1) = -G * AR(M,M-1)
1104.01 AI(M,M-1) = -G * AI(M,M-1)
1105.01 180 CONTINUE
1106.01 C
1107.01 200 RETURN
1108.01 C ::::::::::: last card of CORTH ::::::::::::
1109.01 END

```